Hit List

Search Results - Record(s) 1 through 10 of 16 returned.

☐ 1. Document ID: US 6738821 B1

L4: Entry 1 of 16   File: USPT   May 18, 2004

DOCUMENT-IDENTIFIER: US 6738821 B1
TITLE: Ethernet storage protocol networks

Brief Summary Text (21):
The advantages of the present invention are many and substantial. Most notably, the
Ethernet storage protocol (ESP) of the present invention simplifies the
communication elements needed to transfer data over a network and enables nearly
unlimited scaleablity. The ESP preferably implements a simple transport protocol
(STP) that requires less CPU processing than conventional TCP. It is estimated that
CPU utilization for networks using ESP may be as small as 1/5 of networks using
TCP. In a more preferred embodiment, the ESP will be implemented primarily using
hardware and simple software drivers in order to further limit CPU requirements.
The ESP also preferably takes advantage of a storage encapsulation protocol (SEP)
which is configured to encapsulate portions of storage data, such as SCSI data,
ATAPI data, UDMA data, etc. In communication, senders and targets establish
communication sessions by exchanging handles, which are used to identify the
senders and targets in subsequent communication transactions. Once a session is
open, the session preferably will remain open for the entire time the target and
host are connected to the ESP network. Another advantage of the present invention
is that Ethernet frames are counted to determine whether packets have successfully
been transferred. This is substantially more efficient than prior art techniques
utilizing TCP, which rely on byte counting and complicated time-out calculations.

Drawing Description Text (10):
FIG. 1G illustrates another embodiment in which an ESP network can be connected to
a Fibre Channel storage network.

Drawing Description Text (19):
FIG. 4C illustrates an encapsulation technique for parallel SCSI phases for use on
serial protocols, in accordance with one embodiment of the present invention.

Detailed Description Text (3):
The STP can be configured to transfer any type of data, however, the exemplary
embodiments described herein illustrate the advantages of STP in relation to
storage data transfers that need to be communicated over a network connection at
fast speeds. Exemplary connection speeds are those defined by any of the Ethernet
standards (e.g., 10Base T Ethernet, 100Base T Ethernet, Gigabit Ethernet, 10
Gigabit Ethernet, etc.). Other connection links can also include Fibre Channel.

Detailed Description Text (4):
In one embodiment, the transported data will be in the form of SCSI data, which is
encapsulated for communication using a storage encapsulation protocol (SEP). In
another embodiment, the transported data can be ATAPI data, UDMA data, or any other

h   e b   b g ee e f   e   h   ef   b   e

data that can benefit from the efficient network communication of the ESP of the present invention.

Detailed Description Text (23):
FIG. 1G illustrates yet another embodiment of the present invention. In this embodiment, it is illustrated how an ESP enabled network can be linked to a Fibre Channel storage network. As discussed earlier, a number of desk top computers 121a-121c can be connected to a cluster server system 130 (or simply a single server). The cluster server system 130 preferably has computers that are ESP enabled. Communication from the desk top computers 121a-121c is preferably over a standard Ethernet connection. The cluster server system 130 is in turn connected to a level 2 Ethernet switch 126. The level 2 Ethernet Switch 126 can then be connected to a level 3 Ethernet Switch when IP encapsulates STP. In some environments, it may be desirable to connect the ESP storage network to a Fibre Channel storage network.

Detailed Description Text (24):
As is well known, Fibre Channel uses Fibre Channel Protocol (FCP) to transmit SCSI data. Traditionally, in order to communicate SNMP commands, Fibre Channel networks also use Ethernet cabling to communicate the configuration commands. As shown, the Fibre Channel storage network uses cables 182 to inter connect the network. The cables 182 include Fibre Channel cabling as well as Ethernet cabling. Thus, the SCSI data can be communicated using Fibre Channel and commands can be communicated using Ethernet. In yet another embodiment, Fibre Channel FCP can use IP to communicate control over the same Fibre Channel media. In either case, the Fibre Channel storage network can be connected to the ESP storage network by way of a Fibre Channel/ESP bridge 180. The bridge 180 can therefore translate between STP used in the ESP storage network and FCP used in the Fibre Channel network. In this example, the Fibre Channel storage network includes a cluster server system 130' that has Fibre Channel enabled servers and desk top computers 102d-102e that can communicate in the Fibre Channel environment.

Detailed Description Text (30):
FIG. 3A shows a block diagram of the elements that define an STP header 206, in accordance with one embodiment of the present invention. The STP header 206 will preferably include a handle field 206a, a type field 206b, a length field 206c, a sequence number field 206d, and an acknowledgment field 206e. In one preferred embodiment, the handle field 206a will be defined by a 16-bit number which identifies the endpoint on the destination target or sender (i.e., initiator). Handles are exchanged during a session open, with each end of the transaction supplying a handle for the other end to use for communicating subsequent STP packets 202 during the session. Even if no packets are being exchanged at a particular time during the session, that session will remain open. This is different than Fibre Channel, which is required to open and close a session each time an exchange occurs.

Detailed Description Text (36):
FIG. 4B defines a table that illustrates the SCSI phase definitions as well as the information transferred definitions used in FIG. 4A. FIG. 4C illustrates an encapsulation technique for parallel SCSI phases for use on serial protocols. An SEP header is added to each segment of information to identify the type of information (i.e., which phase it comes from), that the segment carries. In other embodiments, for commonly occurring information pairs such as command plus write data and read data plus status, a single header can be shared by defining a special paired type. Because tag commands are typically used, requiring a tag message phase with each command, status, or data phase, the tag is combined with the header. By using a separate session for LUN (for those devices which have multiple LUNs), the LUN and SCSI ID information is implicit in the choice of session over which the encapsulated SCSI is sent.

Detailed Description Text (52):

h   e b   b g ee e f   e   h   ef   b   e

The simple transport protocol (STP) will now be described with reference to exemplary SCSI transactions. STP provides a low overhead, LAN-oriented transport to efficiently and reliably move blocks of data to and from target devices. As described above, the host computers should have a PIC to enable communication in accordance with the ESP and the target devices should be native ESP devices. Alternatively, the target devices can be off-the-shelf devices that can gain access to the ESP network by way of a bridge circuit 131, as described with reference to FIG. 1C. Also important to note is that STP can either be used alone without IP for some small LAN environments and in other cases STP can be run over IP for larger LAN environments or communications over routers and the Internet. In a preferred embodiment, where SCSI data is being moved, STP will move SCSI data that has been serialized and encapsulated using the described SEP protocol.

| Full | Title | Citation | Front | Review | Classification | Date | Reference | Sequences | Attachments | Claims | KWIC | Draw D |

2. Document ID: US 6683883 B1

L4: Entry 2 of 16   File: USPT   Jan 27, 2004

DOCUMENT-IDENTIFIER: US 6683883 B1
TITLE: ISCSI-FCP gateway

Abstract Text (1):
A method for transferring information between an iSCSI device operating under an iSCSI protocol within a TCP/IP network and a SCSI over Fiber Channel (FCP) device operating under an FCP protocol within an FC network, including: coupling the TCP/IP and the FC networks via a gateway which conveys data between the networks; generating in the gateway a virtual-FC-address for the iSCSI device; generating in the gateway a virtual-TCP/IP-address for the FCP device; conveying a first FC-data-frame, from the FCP device addressed to the virtual-FC-address of the iSCSI device; translating in the gateway the first FCP-data-frame into a first iSCSI-data-frame, addressed to a TCP/IP address of the iSCSI device; and conveying the first iSCSI-data-frame from the gateway to the iSCSI device responsive to the TCP/IP address.

Brief Summary Text (2):
The present invention relates generally to data transfer, and specifically to data transfer between Internet Small Computer System Interface devices and devices operating in a Fibre Channel network.

Brief Summary Text (6):
An Internet SCSI (iSCSI) protocol has been developed by the Internet Engineering Task Force (IETF) to enable SCSI clients and I/O devices to communicate with no limitations on distance between the components. A draft of the protocol can be found at http://ietf.org/internet-drafts/draft-ietf-ips-iscsi-08.txt, and is incorporated herein by reference. The iSCSI protocol encapsulates SCSI commands by representing them as serial strings of bytes preceded by iSCSI headers. The strings of bytes with iSCSI headers, termed Protocol Data Units (PDUs), are formed into TCP/IP packets which are transmitted in a TCP/IP network.

Brief Summary Text (7):
A number of Fibre Channel (FC) protocols have been issued by the American National Standards Institute, Washington, D.C. A Fibre Channel protocol enables transfer of data-frames via an FC switching fabric controlled by a management facility. An FCP protocol is a mapping of the SCSI protocol into an FC protocol, and is a protocol that supports data transfer between hosts and SCSI I/O devices over FC networks.

h   e b   b g ee e f   e   h   ef   b   e

Brief Summary Text (10):
It is an object of some aspects of the present invention to provide a method and apparatus for transferring data between an Internet Small Computer System Interface device and a <u>Fibre Channel</u> device.

Brief Summary Text (11):
In preferred embodiments of the present invention, a gateway couples a network operating under a Transmission Control Protocol/Internet Protocol (TCP/IP) with a network operating under a <u>Fibre Channel</u> (FC) protocol, so as to transfer data between the networks. At least one device in the TCP/IP network is implemented to operate according to an Internet Small Computer System Interface (iSCSI) protocol, such devices herein being termed iSCSI devices. The FC network comprises at least one FC device implemented to operate according to a SCSI protocol, such devices herein being termed FCP devices.

Brief Summary Text (17):
There is therefore provided, according to a preferred embodiment of the present invention, a method for transferring information between an Internet Small Computer System Interface (iSCSI) device operating under an iSCSI protocol within a Transmission Control Protocol/Internet Protocol (TCP/IP) network and a SCSI over <u>Fibre Channel</u> (FCP) device operating under an FCP protocol within a <u>Fibre Channel</u> (FC) network, including: coupling the TCP/IP and the FC networks via a gateway adapted to convey data between the networks; generating in the gateway a virtual-FC-address compatible with the FC protocol for the iSCSI device; generating in the gateway a virtual-TCP/IP-address compatible with the TCP/IP protocol for the FCP device; conveying a first FC-data-frame, compatible with the FCP protocol and comprising FCP-data, from the FCP device addressed to the virtual-FC-address of the iSCSI device; translating in the gateway the first FCP-data-frame into a first iSCSI-data-frame, compatible with the iSCSI protocol and comprising the FCP-data, addressed to a TCP/IP address of the iSCSI device; conveying the first iSCSI-data-frame from the gateway to the iSCSI device responsive to the TCP/IP address; conveying a second iSCSI-data-frame, compatible with the iSCSI protocol and comprising iSCSI-data, from the iSCSI device addressed to the virtual-TCP/IP-address of the FCP device; translating in the gateway the second iSCSI-data-frame into a second FCP-data-frame, compatible with the FCP protocol and comprising the iSCSI-data, addressed to an FC address of the FCP device; and conveying the second FCP-data-frame from the gateway to the FCP device responsive to the FC address.

Brief Summary Text (22):
There is further provided, according to a preferred embodiment of the present invention, apparatus for transferring information between an Internet Small Computer System Interface (iSCSI) device operating under an iSCSI protocol within a Transmission Control Protocol/Internet Protocol (TCP/IP) network and a SCSI over <u>Fibre Channel</u> (FCP) device operating under an FCP protocol within a <u>Fibre Channel</u> FC network, including: a gateway, including a central processing unit (CPU) and a memory, wherein the gateway couples the TCP/IP and the FC networks and is adapted to convey data between the networks, wherein the CPU: generates in the memory a virtual-FC-address compatible with the FC protocol for the iSCSI device; and generates in the memory a virtual-TCP/IP-address compatible with the TCP/IP protocol for the FCP device; and wherein the gateway is adapted to: receive a first FCP-data-frame, compatible with the FCP protocol and comprising FCP-data, from the FCP device addressed to the virtual-FC-address of the iSCSI device; translate the first FCP-data-frame into a first iSCSI-data-frame, compatible with the iSCSI protocol and comprising the FCP-data, addressed to a TCP/IP address of the iSCSI device; convey the first iSCSI-data-frame from the gateway to the iSCSI device responsive to the TCP/IP address; receive a second iSCSI-data-frame, compatible with the iSCSI protocol and comprising iSCSI-data, from the iSCSI device addressed to the virtual-TCP/IP-address of the FCP device; translate the second iSCSI-data-frame into a second FCP-data-frame, compatible with the FCP protocol and comprising

h   e b   b g ee e f   e   h   ef   b   e

the iSCSI-data, addressed to an FC address of the FCP device; and convey the second FCP-data-frame from the gateway to the FCP device responsive to the FC address.

Drawing Description Text (3):
FIG. 1 which is a schematic diagram illustrating transferring data between an Internet Small Computer System Interface (iSCSI) device in a TCP/IP network and a Fibre Channel (FC) device operating as a SCSI device (FCP device) in an FC network;

Detailed Description Text (2):
Reference is now made to FIG. 1, which is a schematic diagram illustrating a system 10 for transferring data between an Internet Small Computer System Interface (iSCSI) device and a Fibre Channel (FC) device implemented to operate as a SCSI device (FCP device). A network 16 operating under a Transmission Control Protocol/Internet Protocol (TCP/IP) comprises one or more generally similar hosts 14, which may be any host which operates according to a SCSI protocol. At least one of hosts 14 is able to operate as an initiator of SCSI commands. At least one of hosts 14 is able to operate as a target of SCSI commands. The terms initiator and target are described in the Background of the Invention; typically, an initiator comprises a computer which uses a target, and a target comprises a storage drive, a printer, or any other input/output device. Each host 14, herein also termed iSCSI devices C, D, E, . . . , comprises a converter 18 which is able to convert SCSI data to iSCSI data, and vice versa. The SCSI data transfers between the converter and its host 14, and the iSCSI data transfers between the converter and TCP/IP network 16, so that the specific host is able to communicate with other entities within the network. Converters which operate as converter 18 are known in the art, and may take the form of software, hardware, or a combination of software and hardware. An example of such a converter is an 1000x1 iSCSI network interface card (NIC) produced by Alacritech Inc., of San Jose, Calif. As described in the Background of the Invention, transmitting SCSI data to and from SCSI devices in the form of iSCSI data enables the SCSI devices to function in network 16. Network 16 also comprises an IP name server 38, which preferably acts as a storage name server.

Detailed Description Text (3):
System 10 comprises a network 32 operating under a Fibre Channel (FC) protocol. FC network 32 comprises one or more SCSI devices 34, which may be any device which operates according to a SCSI protocol and which is able to receive and transmit SCSI data according to an FCP protocol, as described in the Background of the Invention. Devices 34 are herein also termed FCP device A, B, ... At least one device 34 is able to operate as an initiator of SCSI commands. At least one device 34 is able to operate as a target of SCSI commands. Network 32 preferably also comprises an FC Simple Name Server 36.

CLAIMS:

1. A method for transferring information between an Internet Small Computer System Interface (iSCSI) device operating under an iSCSI protocol within a Transmission Control Protocol/Internet Protocol (TCP/IP) network and a SCSI over Fibre Channel (FCP) device operating under an FCP protocol within a Fibre Channel (FC) network, comprising: coupling the TCP/IP and the FC networks via a gateway adapted to convey data between the networks; generating in the gateway a virtual-FC-address compatible with the FC protocol for the iSCSI device; generating in the gateway a virtual-TCP/IP-address compatible with the TCP/IP protocol for the FCP device; conveying a first FC-data-frame, compatible with the FCP protocol and comprising FCP-data, from the FCP device addressed to the virtual-FC-address of the iSCSI device; translating in the gateway the first FCP-data-frame into a first iSCSI-data-frame, compatible with the iSCSI protocol and comprising the FCP-data, addressed to a TCP/IP address of the iSCSI device; conveying the first iSCSI-data-frame from the gateway to the iSCSI device responsive to the TCP/IP address;

h   e b   b g ee e f   e   h   ef   b   e

conveying a second iSCSI-data-frame, compatible with the iSCSI protocol and comprising iSCSI-data, from the iSCSI device addressed to the virtual-TCP/IP-address of the FCP device; translating in the gateway the second iSCSI-data-frame into a second FCP-data-frame, compatible with the FCP protocol and comprising the iSCSI-data, addressed to an FC address of the FCP device; and conveying the second FCP-data-frame from the gateway to the FCP device responsive to the FC address.

7. Apparatus for transferring information between an Internet Small Computer System Interface (iSCSI) device operating under an iSCSI protocol within a Transmission Control Protocol/Internet Protocol (TCP/IP) network and a SCSI over Fibre Channel (FCP) device operating under an FCP protocol within a Fibre Channel FC network, comprising: a gateway, comprising a central processing unit (CPU) and a memory, wherein the gateway couples the TCP/IP and the FC networks and is adapted to convey data between the networks, wherein the CPU: generates in the memory a virtual-FC-address compatible with the FC protocol for the iSCSI device; and generates in the memory a virtual-TCP/IP-address compatible with the TCP/IP protocol for the FCP device; and wherein the gateway is adapted to: receive a first FCP-data-frame, compatible with the FCP protocol and comprising FCP-data, from the FCP device addressed to the virtual-FC-address of the iSCSI device; translate the first FCP-data-frame into a first iSCSI-data-frame, compatible with the iSCSI protocol and comprising the FCP-data, addressed to a TCP/IP address of the iSCSI device; convey the first iSCSI-data-frame from the gateway to the iSCSI device responsive to the TCP/IP address; receive a second iSCSI-data-frame, compatible with the iSCSI protocol and comprising iSCSI-data, from the iSCSI device addressed to the virtual-TCP/IP-address of the FCP device; translate the second iSCSI-data-frame into a second FCP-data-frame, compatible with the FCP protocol and comprising the iSCSI-data, addressed to an FC address of the FCP device; and convey the second FCP-data-frame from the gateway to the FCP device responsive to the FC address.

| Full | Title | Citation | Front | Review | Classification | Date | Reference | Sequences | Attachments | Claims | KWIC | Draw De |

☐   3.   Document ID: US 6591207 B2

L4: Entry 3 of 16   File: USPT   Jul 8, 2003

DOCUMENT-IDENTIFIER: US 6591207 B2
TITLE: Semiconductor production system

Brief Summary Text (4):
The storage area network is an independent network which is constructed of only storages, devices for storing data, by separating the storages from a server. Examples of such storage area networks include those networks based on such links as a fiber channel (one of serial interface standards) described in WO 00/18049 and WO 00/17769 and an optical fiber described in WO 00/2954. The storage area network is a general term for networks that link storage devices independently of the kind of communication devices used. A link of storage devices through a serial bus as defined in IEEE1394 and a link of storage devices through a switched bus as defined by InfiniBand (registered trade name) are storage area networks. However, Ethernet which handles storage protocol, such as iSCSI (registered trade name) and SEP (SCSI Encapsulation Protocol), is the storage area networks.

Drawing Description Text (24):
FIG. 23 is a block diagram showing a semiconductor inspection apparatus when a

h   e b   b g ee e f   e   h   ef   b   e

fiber channel is employed as the storage area network.

Drawing Description Text (29):
FIG. 28 is a diagram showing a semiconductor manufacturing apparatus when a fiber channel is used as the storage area network and dedicated hardware is mounted in its control unit.

Detailed Description Text (4):
FIG. 2 is a block diagram illustrating another configuration of the semiconductor production system. As shown in the figure, the storage area network 40 interconnects a semiconductor inspection apparatus 10, a semiconductor manufacturing apparatus 20 and a plurality of storage devices 30. The storage area network 40 employs fiber channels 41 as communication devices and interconnects the fiber channels with a fabric 42. The semiconductor inspection apparatus 10 and the semiconductor manufacturing apparatus 20 are linked together via the general network 50.

Detailed Description Text (6):
By connecting the fabrics using fiber channels, it is possible to arbitrarily select installation locations of the semiconductor inspection apparatus 10, the semiconductor manufacturing apparatus 20 and the storage devices 30. For example, the semiconductor inspection apparatus 10 and the semiconductor manufacturing apparatus 20 may be installed at a manufacturing site and the storage device 30 at an office or data center. With this arrangement, if the manufacturing site is hit by a natural disaster, because all the information associated with the semiconductor inspection apparatus 10 and the semiconductor manufacturing apparatus 20 installed at the manufacturing site is stored in the storage device 30, the recovery from damages will be easy. The configuration using switches in the fabric 42 is identical to those using the InfiniBand for the communication device. Hence, where there is no need for a long-distance communication through the fiber channels 41, the use of the InfiniBand can realize a compact system of the identical configuration.

Detailed Description Text (7):
FIG. 3 is a block diagram showing still another configuration of the semiconductor production system. As shown in the figure, the storage area network 40 interconnects a semiconductor inspection apparatus 10, a semiconductor manufacturing apparatus 20 and a plurality of storage devices 30. The storage area network 40 adopts fiber channels 41 as the communication device connecting the individual apparatus in loop. The semiconductor inspection apparatus 10 and the semiconductor manufacturing apparatus 20 are interconnected through the general network 50. This loop configuration does not require facilities such as fabrics but realizes a simple system that can be built only by connecting fiber optics. This configuration facilitates maintenance and can also achieve a system with duplicated loops easily.

Detailed Description Text (32):
FIG. 23 is a block diagram showing a semiconductor inspection apparatus when a fiber channel is employed as the storage area network. As shown in the figure, inspection apparatus 10 comprises an observation unit 12 having an optical image sensing device and others and a control unit 14 for controlling the observation unit. The observation unit 12 and the control unit 14 are connected through the storage area network 40 to a storage device 30 outside the inspection apparatus, a storage device 31 inside the inspection apparatus, and a plurality of inspection processing apparatus 60. The storage area network 40 has a plurality of fabrics 42, to each of which the units and apparatus are connected via fiber channels 41. The fabrics 42 are interconnected also by a fiber channel 43. In this case, when the connections 43 between a plurality of fabrics are replaced with WAN such as ATM, the inspection apparatus may be installed in a clean room at the manufacturing site and the inspection processing apparatus in a remote office.

h   e b   b g ee e f   e   h   ef   b   e

Detailed Description Text (37):
FIG. 28 is a block diagram showing a semiconductor manufacturing apparatus when a fiber channel is adopted for the storage area network and a dedicated hardware is used for the control unit. As shown in the figure, the manufacturing apparatus 20 comprises an exposure unit 22 having an optical exposure means or charged particle exposure means, and a control unit 24 for controlling the exposure unit. The exposure unit 22 and the control unit 24 are connected through the storage area network 40 to a storage device 30 outside the manufacturing apparatus, a storage device 32 inside the manufacturing apparatus, and a plurality of design information processing apparatus 60. The storage area network 40 has a plurality of fabrics 42, to which various units and apparatus are connected by fiber channels 41. The fabrics 42 are interconnected also by the fiber channel 43. The control unit 24 comprises a BM (buffer memory) 25 for temporarily storing design information, a recovery unit 26 for processing the design information for use in the control unit 24, a dividing unit 27 for dividing the data processed by the recovery unit 26 into minimum geometric units such as rectangles, a proximity correction unit 28 for executing a proximity effect correction on the minimum geometric units divided by the dividing unit, and a shot unit 29 for converting the data into shot information conforming to the exposure unit 22.

Detailed Description Text (38):
The conventional control unit 24 implemented with hardware is a black box whose inner data cannot be accessed. With this embodiment, however, the data contained in the control unit 24 can be accessed via the fiber channels. Hence, the correction result produced by the proximity correction unit 28 and the actual shot specification information produced by the shot unit 29 can be retrieved and, based on these information, new functions can be provided.

Detailed Description Text (46):
(2) The semiconductor production system according to item (1), wherein the storage area network has a plurality of fabrics for switching fiber channels.

CLAIMS:

2. The semiconductor production system according to claim 1, wherein the storage area network has a plurality of fabrics for switching fiber channels.

| Full | Title | Citation | Front | Review | Classification | Date | Reference | Sequences | Attachments | Claims | KWIC | Draw De |

☐ 4. Document ID: US 6493750 B1

L4: Entry 4 of 16   File: USPT   Dec 10, 2002

DOCUMENT-IDENTIFIER: US 6493750 B1
TITLE: Command forwarding: a method for optimizing I/O latency and throughput in fibre channel client/server/target mass storage architectures

Brief Summary Text (4):
The fibre channel ("FC") is an architecture and protocol for a data communication network for interconnecting a number of different combinations of computers and peripheral devices. The FC supports a variety of upper-level protocols, including the small computer systems interface ("SCSI") protocol. A computer or peripheral

h   e b   b g ee e f   e   h   ef   b   e

device is linked to the network through an FC port and copper wires or optical
fibres. An FC port includes a transceiver and an interface controller, and the
computer peripheral device in which the FC port is contained is called a "host."
The FC port exchanges data with the host via a local data bus, such as a peripheral
computer interface ("PCI") bus. The interface controller conducts lower-level
protocol exchanges between the fibre channel and the computer or peripheral device
in which the FC port resides.

Brief Summary Text (7):
With the advent of the FC, client computers, server computers, and mass storage
devices may all be symmetrically interconnected by a single communications medium.
The traditional client/server architecture is commonly ported to the FC using the
same type of client/server protocols as are used in the LAN and SCSI networks
discussed above. Implementing a client/server architecture using these
client/server protocols introduces significant inefficiencies in data transfer with
concomitant increases in I/O latency and loss of fibre channel throughput.
Designers and manufacturers of fibre channel ports that are used to interconnect
mass storage devices, client computers and server computers, have recognized the
need for designing new client/server protocols that are better suited to take
advantage of the FC.

Brief Summary Text (9):
The present invention provides a fibre channel ("FC") interface controller that
supports, in hardware, efficient client/server data transfer protocols, and
provides a method for implementing the efficient data transfer protocols. The
interface controller of the present invention supports the forwarding of a read or
write command from a server computer to a mass storage device such that the
identification contained in the forwarded data transfer command is that of the
client computer, rather than the server computer that forwards the data transfer
command. The interface controller also supports copying of the source
identification contained in data and status messages sent by a mass storage device
to a client computer in response to the data transfer command automatically into
the data structure that is used by the interface controller to check correspondence
between the destination identification associated with a data transfer command
issued by the client computer and the source identification of data messages and
status messages received back by the client computer in response to the data
transfer command. These two features of the interface controller enable the data
transfer protocol of the present invention. In the data transfer protocol, a server
computer, in response to receiving a read command from a client computer, forwards
the read command to a target mass storage device, which, in turn, transmits the
data requested by the read command and a final status message directly back to the
requesting client computer. In response to a write command received by a server
computer from a client computer, the server computer forwards the write command to
a target mass storage device. The target mass storage device sends a transfer ready
status message directly to the client computer, receives, directly from the client
computer, the data to be written to the mass storage device, and finally sends a
status message directly back to the client computer. This improved data transfer
protocol eliminates unnecessary storing and forwarding of data by the server
computer and eliminates the transfer of redundant data messages and status messages
within the communications network.

Drawing Description Text (11):
FIG. 10 shows the basic underlying circular queue data structure used in the
TachLite Fibre Channel Mass Storage Interface Controller interface.

Drawing Description Text (24):
FIG. 23 illustrates a client/server architecture implemented on top of the fibre
channel.

Drawing Description Text (25):

h   e b   b g ee e f   e   h   ef   b   e

FIG. 24 illustrates implementation of a read operation according to currently-available client/server protocols mapped onto the <u>fibre channel</u>.

<u>Detailed Description Text</u> (2):
The present invention will be described below in six subsections. The first three subsections provide greater detail about the <u>fibre channel</u> architecture and protocol, the SCSI architecture and protocol, and implementation of the SCSI protocol on top of the <u>fibre channel</u> protocol. The fourth subsection discusses the <u>fibre channel</u> arbitrated loop intialization process. The fifth subsection provides a general description of the present invention, and the sixth subsection provides a detailed description of an implementation of the present invention.

<u>Detailed Description Text</u> (3):
<u>Fibre Channel</u>

<u>Detailed Description Text</u> (4):
The <u>Fibre Channel</u> ("FC") is defined by, and described in, a number of ANSI Standards documents, including: (1) <u>Fibre Channel</u> Physical and Signaling Interface ("FC-PH"), ANSI X3.230-1994, ("FC-PH-2), ANSI X3.297-1997; (2) <u>Fibre Channel</u>-- Arbitrated Loop ("FC-AL-2"), ANSI X3.272-1996; (3) <u>Fibre Channel</u>--Private Loop SCSI Direct Attached ("FC-PLDA"); (4) <u>Fibre Channel</u>--Fabric Loop Attachment ("FC-FLA"); (5) <u>Fibre Channel</u> Protocol for SCSI ("FCP"); (6) <u>Fibre Channel</u> Fabric Requirements ("FC-FG"), ANSI X3.289:1996; and (7) <u>Fibre Channel</u> 10-Bit Interface. These standards documents are under frequent revision. Additional <u>Fibre Channel</u> System Initiative ("FCSI") standards documents include: (1) Gigabaud Link Module Family ("GLM"), FCSI-301; (2) Common FC-PH Feature Sets Profiles, FCSI-101; and (3) SCSI Profile, FCSI-201. These documents may be found at the world wide web Internet page having the following address:

<u>Detailed Description Text</u> (7):
In the following discussion, "FC" is used as an adjective to refer to the general <u>Fibre Channel</u> architecture and protocol, and is used as a noun to refer to an instance of a <u>Fibre Channel</u> communications medium. Thus, an FC (architecture and protocol) port may receive an FC (architecture and protocol) sequence from the FC (communications medium).

<u>Detailed Description Text</u> (14):
The FC is a full duplex data transmission medium. Frames and sequences can be simultaneously passed in both directions between an originator, or initiator, and a responder, or target. An exchange comprises all sequences, and frames within the sequences, exchanged between an originator and a responder during a single I/O transaction, such as a read I/O transaction or a write I/O transaction. The FC protocol is designed to transfer data according to any number of higher-level data exchange protocols, including the Internet protocol ("IP"), the Small Computer Systems Interface ("SCSI") protocol, the High Performance Parallel Interface ("HIPPI"), and the Intelligent Peripheral Interface ("IPI"). The SCSI bus architecture will be discussed in the following subsection, and much of the subsequent discussion in this and remaining subsections will focus on the SCSI protocol embedded within the FC protocol. The standard adaptation of SCSI protocol to <u>fibre channel</u> is subsequently referred to in this document as "FCP.". Thus, the FC can support a master-slave type communications paradigm that is characteristic of the SCSI bus and other peripheral interconnection buses, as well as the relatively open and unstructured communication protocols such as those used to implement the Internet. The SCSI bus architecture concepts of an initiator and target are carried forward in the FCP, designed, as noted above, to encapsulate SCSI commands and data exchanges for transport through the FC.

<u>Detailed Description Text</u> (16):
The next high level section 308, called the data payload, contains the actual data packaged within the FC frame. The data payload contains data and <u>encapsulating</u>

h　　　e　b　　　　b　g　ee　e　f　　　　　e　　h　　　ef　　b　　　e

protocol information that is being transferred according to a higher-level protocol, such as IP and SCSI. FIG. 3 shows four basic types of data payload layouts 326-329 used for data transfer according to the SCSI protocol. The first of these formats 326, called the FCP_CMND, is used to send a SCSI command from an initiator to a target. The FCP_LUN field 330 comprises an 8-byte address that may, in certain implementations, specify a particular SCSI-bus adapter, a target device associated with that SCSI-bus adapter, and a logical unit number ("LUN") corresponding to a logical device associated with the specified target SCSI device that together represent the target for the FCP_CMND. In other implementations, the FCP_LUN field 330 contains an index or reference number that can be used by the target FC host adapter to determine the SCSI-bus adapter, a target device associated with that SCSI-bus adapter, and a LUN corresponding to a logical device associated with the specified target SCSI device. An actual SCSI command, such as a SCSI read or write I/O command, is contained within the 16-byte field FCP_CDB 332.

Detailed Description Text (40):
The Tachyon TL FC Mass Storage Interface Controller ("TL") is a high-performance, low-cost, loop-based interface controller for use in the FC ports that interconnect peripheral devices and computers to an FC arbitrated loop. In this subsection, an overview of the functionality of, and interface to, the TL will be presented. A more detailed description of the TL is included in U.S. Patent Application Ser. No. 09/183,865, entitled "Fibre Channel Controller," assigned to the Hewlett Packard Company, and filed on Oct. 30, 1998 that is hereby incorporated by reference in its entirety.

Detailed Description Text (45):
The completion message manager 925 manages the IMQ and provides queue entries to the inbound data manager 924 into which the inbound data manager places completion messages. The single frame manager 926 manages the SFQ in host memory and provides entries to the fibre channel services component 927 into which the fibre channel component services place inbound frames. The exchange request manager 931 fetches new entries from the ERQ and sends them to the SCSI exchange manger-outbound ("SEM-OUT") for processing. The inbound data manager 924 informs the inbound frame processors, i.e. the SCSI exchange manager-inbound ("SEM-IN") 928 and fibre channel services component 927, of new frames and routes the frames to their proper destination in the host. Also, the inbound data manager sends completion messages to the host via the IMQ. The fibre channel services component 927 manages the fibre channel frames that the SEM-IN 928 does not manage. The fibre channel services component places the frames in the SFQ. The SEM-IN 928 manages the phases of a SCSI exchange that receive a fibre channel sequence. The SEM-IN reads the SEST entries via the SEST link fetch manager 929 and either sends the inbound data to the proper host buffers or sends the request to the SEM-OUT 932 to send the next phases of fibre channel sequence. The SEST link fetch manager 929 is responsible for reading and writing SEST entries, depending upon requests from the SEM-IN 928 and SEMOUT 932 components. The SEM-OUT 932 manages the phases of a SCSI exchange that require a fibre channel sequence to be sent. The SEM-OUT 932 reads the SEST entries via the SEST link fetch manager 929, builds the request to send those sequences, and sends the requests to the outbound sequence manager 930. The outbound sequence manager ("OSM") 930 processes requests from the SEM-OUT 932 to send fibre channel sequences from the host and retrieves fibre channel frame headers and payloads from the host to send to the remote node. The OSM segments the sequence into fibre channel frames of up to 1 KByte in size and queues them into the outbound FIFO manager 916.

Detailed Description Text (48):
FIG. 11 shows a more detailed view of the host memory data structures required to perform an FCP write operation where the FC node in which a TL resides is the initiator of the FCP write operation and where the data payload that will include the data to be written requires 4 or more data buffers. The host prepares an initiator write entry ("IWE") 1102 within a SEST entry 1104 in the SEST (904 in FIG. 9). Associated with the IWE are: (1) a fibre channel header structure ("FCHS")

h   e   b   b   g   ee e f   e   h   ef   b   e

1106 that is used to send the FCP DATA sequence; (2) a data buffer 1108 that is used to receive the FCP_RSP frame from the SCSI target; and (3) one or more extended scatter gather list ("SGL") pages 1110 and 1112 that contain pointers to data buffers 1113-1117 in which the host places the data to be written to the SCSI target via the FCP_DATA sequence. The host then creates an I/O request block ("IRB") 1118 in an unused ERQ entry 1120 and associates the IRB with an FCHS 1122 that is used for the FCP_CMND sequence. The host then increments the ERQ producer index. The producer index increment is detected by the TL, and the TL then launches the FCP write operation. The TL uses the information and data stored within the IRB 1118 and RWE 1102, and the data structures associated with the IRB and RWE, to conduct the entire FCP write operation, including the FCP_CMND sequence, the FCP_XFER_RDY sequence, and the FCP_DATA sequence. The TL receives from the target a FCP_RSP sequence at the completion of the FCP write operation.

Detailed Description Text (69):
Unfortunately, the elimination of a need for the bridging function within the server by the FC is not currently taken advantage of by client/server protocols implemented on top of the FC. FIG. 24 illustrates implementation of a read operation according to currently-available client/server protocols mapped onto the FC. The client computer 2402 is shown in FIG. 24 with ID equal to (A,) or, in other words, having the fibre channel address "A." Similarly, the server computer 2404 has ID equal to "B," and the mass storage device 2406 has ID equal to "C." The read command, data, and a status message are exchanged between the client computer 2402, the server computer 2404, and the mass storage device 2406 within a set of single-frame and possibly multi-frame FCP sequences 2407-2414. In FIG. 24, each FCP sequence or FC frame contains an identification of the FC sequence type as well as the S_ID of the FC node (the client computer 2402, the server computer 2404, or mass storage device 2406) that sends the FCP sequence. For example, FCP sequence 2407 is an FCP_CMND with SJID equal to "A." These illustration conventions will also be used in FIGS. 25-26.

CLAIMS:

5. The interface controller of claim 1 wherein the communications network is a fibre channel communications network.

| Full | Title | Citation | Front | Review | Classification | Date | Reference | Sequences | Attachments | Claims | KWIC | Draw De |

☐ 5. Document ID: US 6477139 B1

L4: Entry 5 of 16   File: USPT   Nov 5, 2002

DOCUMENT-IDENTIFIER: US 6477139 B1
TITLE: Peer controller management in a dual controller fibre channel storage enclosure

Brief Summary Text (4):
The fibre channel ("FC") is an architecture and protocol for a data communication network for interconnecting a number of different combinations of computers and peripheral devices. The FC supports a variety of upper-level protocols, including the small computer systems interface ("SCSI") protocol. A computer or peripheral device is linked to the network through an FC port and copper wires or optical fibres. An FC port includes a transceiver and an interface controller, and the

h   e b   b g ee e f   e   h   ef   b   e

computer peripheral device in which the FC port is contained is called a "host."
The FC port exchanges data with the host via a local data bus, such as a peripheral
computer interface ("PCI") bus. The interface controller conducts lower-level
protocol exchanges between the fibre channel and the computer or peripheral device
in which the FC port resides.

Drawing Description Text (15):
FIG. 13B illustrates the connection of a disk drive to a fibre channel loop via a
port bypass circuit.

Detailed Description Text (2):
The present invention will be described below in six subsections. The first three
subsections provide greater detail about the fibre channel architecture and
protocol, the SCSI architecture and protocol, and implementation of the SCSI
protocol on top of the fibre channel protocol. The fourth subsection discusses the
fibre channel arbitrated loop initialization process. The fifth subsection provides
a general description of multi-peripheral-device enclosures, and the sixth
subsection describes a specialized SCSI command set and protocol used for component
management within systems of peripheral devices that communicate with one or more
host computers via the SCSI protocol. The seventh subsection provides a detailed
description of a hardware embodiment of the present invention, and a final eighth
subsection provides a detailed description of inter-processor communications and
control sharing within a multi-peripheral-device enclosure.

Detailed Description Text (3):
Fibre Channel

Detailed Description Text (4):
The Fibre Channel ("FC") is defined by, and described in, a number of ANSI
Standards documents, including: (1) Fibre Channel Physical and Signaling Interface
("FC-PH"), ANSI X3.230-1994, ("FC-PH-2"), ANSI X3.297-1997; (2) Fibre Channel-
Arbitrated Loop ("FC-AL-2"), ANSI X3.272-1996; (3) Fibre Channel-Private Loop SCSI
Direct Attached ("FC-PLDA"); (4) Fibre Channel-Fabric Loop Attachment ("FC-FLA");
(5) Fibre Channel Protocol for SCSI ("FCP"); (6) Fibre Channel Fabric Requirements
("FC-FG"), ANSI X3.289:1996; and (7) Fibre Channel 10-Bit Interface. These
standards documents are under frequent revision. Additional Fibre Channel System
Initiative ("FCSI") standards documents include: (1) Gigabaud Link Module Family
("GLM"), FCSI-301; (2) Common FC-PH Feature Sets Profiles, FCSI-101; and (3) SCSI
Profile, FCSI-201. These documents may be found at the world wide web Internet page
having the following address: "http://www.fibrechannel.com"

Detailed Description Text (7):
In the following discussion, "FC" is used as an adjective to refer to the general
Fibre Channel architecture and protocol, and is used as a noun to refer to an
instance of a Fibre Channel communications medium. Thus, an FC (architecture and
protocol) port may receive an FC (architecture and protocol) sequence from the FC
(communications medium).

Detailed Description Text (14):
The FC is a full duplex data transmission medium. Frames and sequences can be
simultaneously passed in both directions between an originator, or initiator, and a
responder, or target. An exchange comprises all sequences, and frames within the
sequences, exchanged between an originator and a responder during a single I/O
transaction, such as a read I/O transaction or a write I/O transaction. The FC
protocol is designed to transfer data according to any number of higher-level data
exchange protocols, including the Internet protocol ("IP"), the Small Computer
Systems Interface ("SCSI") protocol, the High Performance Parallel Interface
("HIPPI"), and the Intelligent Peripheral Interface ("IPI"). The SCSI bus
architecture will be discussed in the following subsection, and much of the
subsequent discussion in this and remaining subsections will focus on the SCSI

h   e b   b g ee e f   e   h   ef   b   e

protocol embedded within the FC protocol. The standard adaptation of SCSI protocol to fibre channel is subsequently referred to in this document as "FCP." Thus, the FC can support a master-slave type communications paradigm that is characteristic of the SCSI bus and other peripheral interconnection buses, as well as the relatively open and unstructured communication protocols such as those used to implement the Internet. The SCSI bus architecture concepts of an initiator and target are carried forward in the FCP, designed, as noted above, to encapsulate SCSI commands and data exchanges for transport through the FC.
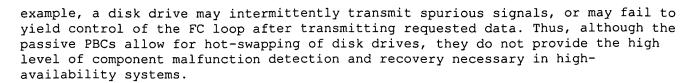
Detailed Description Text (16):
The next high level section 308, called the data payload, contains the actual data packaged within the FC frame. The data payload contains data and encapsulating protocol information that is being transferred according to a higher-level protocol, such as IP and SCSI. FIG. 3 shows four basic types of data payload layouts 326-329 used for data transfer according to the SCSI protocol. The first of these formats 326, called the FCP_CMND, is used to send a SCSI command from an initiator to a target. The FCP_LUN field 330 comprises an 8-byte address that may, in certain implementations, specify a particular SCSI-bus adapter, a target device associated with that SCSI-bus adapter, and a logical unit number ("LUN") corresponding to a logical device associated with the specified target SCSI device that together represent the target for the FCP_CMND. In other implementations, the FCP_LUN field 330 contains an index or reference number that can be used by the target FC host adapter to determine the SCSI-bus adapter, a target device associated with that SCSI-bus adapter, and a LUN corresponding to a logical device associated with the specified target SCSI device. An actual SCSI command, such as a SCSI read or write I/O command, is contained within the 16-byte field FCP_CDB 332.

Detailed Description Text (72):
A single GBIC (for example, GBIC 1216) allows connection of the enclosure to a host computer via an optical fibre. A second GBIC (for example, GBIC 1218) may allow an enclosure to be daisy-chained to another enclosure, thereby adding another group of disk drives to the fibre channel loop 1230. When a second GBIC is present, and no further enclosures are to be daisy-chained through the second GBIC, a loop-back connector, or terminator, is normally plugged into the second GBIC to cause FC signals to loop back through the enclosure and, ultimately, back to the host computer.

Detailed Description Text (74):
FIG. 13B illustrates the connection of a disk drive to a fibre channel loop via a port bypass circuit. In the interest of brevity, the components of the port bypass circuit in 13B that are the same as components shown in FIG. 13A will be labeled in 13B with the same labels used in FIG. 13A, and descriptions of these components will not be repeated. The disk drive 1322 receives an input signal IN 1302 from the fibre channel loop via the Port signal 1312. When the disk drive asserts the SEL control signal 1320, the disk drive provides the signal Pin 1316 that is passed by the multiplexer 1314 to the output signal OUT 1318 that is transmitted via the FC loop to the next FC port in the direction of the FC signal. When the SEL control signal 1320 is de-asserted, the disk drive 1322 is bypassed, and the input signal IN 1302 is passed as the output signal OUT 1318 to the next FC port in the direction of the FC signal. The disk drive 1322 asserts the SEL control signal when it is securely mounted in the enclosure, connected to the backplane, and functionally ready to inter-operate with the FC loop. When the disk drive 1322 is absent, or not functionally ready to inter-operate with the FC loop, the SEL control line 1320 is de-asserted and the FC signal bypasses the disk drive. When the disk drive is hot-swapped into or out of an on-line enclosure, the FC loop that interconnects the functioning disk drives must undergo re-initialization, as discussed above, but the ensuing interruption is relatively slight, and any interrupted data transfers are recovered. However, there are different possible failure modes of disk drives that can degrade or disable operation of the FC loop and that cannot be detected and bypassed by the essentially passive PBC. For

h   e b   b g ee e f   e   h   ef   b   e

example, a disk drive may intermittently transmit spurious signals, or may fail to yield control of the FC loop after transmitting requested data. Thus, although the passive PBCs allow for hot-swapping of disk drives, they do not provide the high level of component malfunction detection and recovery necessary in high-availability systems.

CLAIMS:

13. The highly available, fault-tolerant multi-device enclosure of claim 4 wherein the internal communication medium and the external communications medium are both portions of a <u>fibre channel</u> arbitrated loop.

| Full | Title | Citation | Front | Review | Classification | Date | Reference | Sequences | Attachments | Claims | KWIC | Draw De |

☐ 6. Document ID: US 6470397 B1

L4: Entry 6 of 16   File: USPT   Oct 22, 2002

DOCUMENT-IDENTIFIER: US 6470397 B1
TITLE: Systems and methods for network and I/O device drivers

<u>Brief Summary Text</u> (12):
Many conventional computer systems include at least two interface cards, an I/O HBA and a network interface card, to respectively handle I/O and network protocols. As described above, these cards are managed respectively by a SCSI miniport and an NDIS miniport. Each card is typically connected to separate communication media. For example, the I/O HBA might be connected to a high speed <u>Fibre Channel</u> communication link, which in turn might be connected to one or more server systems or other computer systems. The <u>Fibre Channel</u> link would then handle storage related communications. The NDIS miniport might be connected via a NIC to an Ethernet link, which in turn may be connected to other computer systems. The Ethernet link would then handle the network related communications, such as, in the case of a clustered server system, "heartbeat" information. The network heartbeat traffic is typically much less than the storage related traffic, and therefore the lower bandwidth, lower cost, Ethernet link is considered well suited to handle such lower bandwidth communication. Thus, for example, in a typical clustered server system having host systems and storage subsystems, each host system would have at least one I/O HBA and one Ethernet interface card. In addition, each storage subsystem would have at least one I/O HBA. The host systems communicate heartbeat information to each other over the Ethernet link. Both the host systems and the storage subsystems communicate storage information over the <u>Fibre Channel</u>.

<u>Brief Summary Text</u> (15):
One embodiment of the present invention advantageously enables both network protocol and an I/O protocol to be transferred to a communication link using the same interface circuit. For example, in one embodiment two or more computer systems are interconnected using a communication link, such as a <u>Fibre Channel</u> link. Both an I/O protocol, such as a SCSI protocol, and a network protocol, such as an Internet protocol (IP), are used to transmit data over the same <u>Fibre Channel</u> link using the same host bus adapter. Thus, in comparison with conventional systems, one embodiment of the present requires half as many communication links and half as many host bus adapters to carry data using both SCSI and IP protocols.

h   e b   b g ee e f   e   h   ef   b   e

Detailed Description Text (6):
In one embodiment of the present invention, the cluster storage protocol is a SCSI
protocol, such as SCSI II, while the network protocol is the Internet Protocol
(IP). In one embodiment, the communication link 110 connecting the various cluster
systems or "nodes" is a Fibre Channel Loop (FCL). Fibre Channel is a high-speed
data transfer interface technology that advantageously maps common transport
protocols, such as SCSI and IP. Thus, using Fibre Channel technology, it is
possible to merge high-speed I/O, such as SCSI, and networking functionality in a
single connectivity technology. However, alternative embodiments can use other bus
technologies, such as a SCSI bus, to run both I/O and networking protocols on a
common link. Thus, in one embodiment, the network and storage packets are
transferred between computer systems using standard network and I/O protocols, such
as the IP and SCSI protocols. This embodiment may be used when the HBA, such as one
based on the QLogic ISP2200, supports both IP and SCSI protocols. In another
embodiment, if the HBA supports the SCSI protocol, but not the IP protocol, the
network packets are encapsulated in SCSI packets or commands. In the
"encapsulation" embodiment, the HBA may support the SCSI target mode, as well as
the more typical initiator mode, thereby allowing the HBA to receive SCSI packets
encapsulating IP packets.

Detailed Description Text (7):
The Fibre Channel Loop 110 is connected to the host bus adapter of the host systems
102, 104 and the storage subsystems 106, 108. Thus, storage and network packets can
be routed between the host systems 102, 104 and the storage subsystems 106, 108.
The data transferred using the IP protocol may include "heartbeat" related
information. "Heartbeats" are transferred across the clustered nodes to ensure that
everything is properly synchronized, and so that each node can ensure that the
other nodes are functioning.

Detailed Description Text (8):
The Fibre Channel Loop 110 is connected to the host bus adapters residing in each
node 102-108. Thus, data is transferred from and to the Fibre Channel Loop 110 via
respective host bus adapters. The operating systems running each node communicate
with their respective host bus adapters using device drivers. In one embodiment,
the device driver is responsible for accessing the host bus adapter's internal
registers, starting and completing I/O operations, performing error processing, and
servicing interrupts generated by the device.

Detailed Description Text (9):
In one embodiment, a fail-over path is provided to ensure high reliability and to
avoid a shutdown of the clustered sever system in the event of a communication link
failure. By way of example, FIG. 2 illustrates a clustered server system 200 which
has two Fibre Channel Loops 210, 212. In normal operation, Loop 1210 handles the
I/O traffic, while Loop 2212 handles the network traffic. Thus, both communication
loops 210, 212 are efficiently utilized in normal operation, with the bandwidth of
Loop 1 allocated to I/O traffic, and the bandwidth of Loop 2 allocated to network
traffic. However, in the event of a Fibre Channel Loop failure, such as the failure
of Loop 1210, one embodiment of the present invention detects the failure. Upon
detecting the failure, the remaining operating Fibre Channel Loop, such as Loop
2212, is used as a fail-over path to carry both network traffic and I/O traffic.
Thus, in contrast to conventional systems, which would utilize four communication
links to provide redundancy for both network and I/O traffic, the illustrated
embodiment 200 of the present invention utilizes two communication links 210, 212.
Furthermore, in contrast to conventional systems, which would utilize at least two
HBAs and two NICs to provide such redundancy, the illustrated embodiment 200 uses
two HBAs per host system 202, 204 to provide the desired fail-over path.

Detailed Description Text (10):
In one embodiment, the fail-over technique operates as follows. Typically, a
Windows NT Cluster Administrator is used to manage and configure network paths and

h   e b   b g e e e f   e   h   e f   b   e

storage devices. In conventional systems, if multiple network paths, with corresponding multiple communication links, have been configured, one path is designated as the primary network path. If the Cluster Administrator determines, via a loss of signal indication or the like, that it is unable to communicate over the primary path, the Windows NT Cluster Administrator can switch to a secondary network path and a corresponding secondary communication link. In contrast to conventional systems, one embodiment of the present invention permits the I/O bus, such as the SCSI bus or the Fibre Channel loop, to be configured as a secondary or primary network path in addition to acting as a I/O or storage path. Thus, upon detection of a network path failure, the Windows NT Cluster Administrator can fail-over to the I/O bus.

Detailed Description Text (13):
The software architecture and operation of the embodiment 100 illustrated in FIG. 1 will now be discussed in detail. FIG. 5 illustrates one embodiment of a driver architecture 500 of the present invention. For example, the illustrated embodiment 500 could be used in one of the host systems illustrated in FIGS. 1 or 2. The upper layers of the NIC driver and the SCSI driver are similar to those found in the conventional driver architectures 300, 400 illustrated in FIGS. 3 and 4. However, in contrast to the NIC miniport 302 illustrated in FIG. 3, which interfaces to its own network interface card 312, the NDIS miniport 502 illustrated in FIG. 5 redirects communication to an enhanced SCSI miniport driver 508 via an interface 504. The enhanced SCSI miniport driver 508, in turn, routes network traffic from the NDIS miniport 502 onto the common Fibre Channel Loop 110 illustrated in FIG. 1. Therefore, in one embodiment, the network miniport driver 502 and the SCSI miniport driver 508 allow host-to-host network traffic to be overlaid onto a Fibre Channel bus, or, in an alternative embodiment, onto a SCSI bus. This permits the host-to-host traffic to operate in parallel with normal disk storage related traffic on the same bus. Thus, the enhanced driver 502, 508, in combination with the HBA, emulate the network interface card (NIC).

Detailed Description Text (14):
Thus, in one embodiment, the NDIS miniport driver 502 emulates an Ethernet connection between multiple systems sharing a common Fibre Channel or SCSI bus. This Ethernet emulation is overlaid by the enhanced SCSI miniport onto the same data channels used to access disk devices shared between the clustered systems. As previously described with reference to FIG. 1, this approach particularly benefits clustered systems that are running short on card slots, such as PCI slots. In addition, as previously described with reference to FIG. 2, in one embodiment, the present invention can provide additional fail-over paths to backup NIC adapters used for local host-to-host communications. In one embodiment, the network driver is adapter-type independent and supports multiple Ethernet emulations through multiple enhanced miniport drivers simultaneously.

Detailed Description Text (58):
The embodiment where network and storage packets are transferred between computer systems using standard IP and SCSI protocols, rather than by encapsulating IP packets in SCSI packets, will now be described. The data structures and routines for this embodiment are similar to the data structures and routines for the "encapsulation" embodiment described above, with the following modifications. In general, the modifications enable the NDIS miniport driver 502 and the HBA processor to send network unencapsulated IP packets and the associated scatter/gather lists to be sent to the HBA processor. Furthermore, the modifications allow the receive buffers to be immediately pushed onto the HBA processor receive buffer queue, rather than having the enhanced SCSI miniport 508 get the receive buffers from the enhanced NDIS miniport 502 when requested. The modifications help take advantage of HBAs that support both network and I/O protocols.

Detailed Description Text (63):

h   e b   b g ee e f   e   h   ef   b   e

As illustrated below, the BD_INQUIRY_DATA data structure is modified to export additional information from the enhanced SCSI miniport driver 508 to the NDIS miniport driver 502. For example, the data structure includes a pointer to the receive buffer queue, as well as the size of the receive buffer queue, which can be passed to the NDIS miniport driver 502. This allows the NDIS miniport driver 502 to directly pass the free receive buffers to the HBA processor. Furthermore, an Add Buffers backdoor routine entry is also passed to the NDIS miniport driver 502. The Add Buffers backdoor routine permits the NDIS miniport driver 502 to inform the SCSI miniport driver 508 if free buffers have been added to the receive buffer queue. In addition, the data structure also includes an entry for passing a Fibre Channel adapter world wide node name. Thus, rather than encapsulating the physical address in the network address, as described above, the node name may be used.

Detailed Description Text (67):
The Port Database routine may be modified to keep a separate list of all IP type devices, such as other HBAs attached on the same Fibre loop. This list is used by the enhanced SCSI miniport 508 to get the Fibre Channel address when sending a packet. Because all Fibre Channel addressing is resolved by the SCSI miniport 508, the NDIS miniport 502 no longer uses the port database.

CLAIMS:

3. The system for sending both network and storage protocols over the same host adapter as defined in claim 1, wherein said communication medium link is fibre channel-compatible.

7. The system for sending both network and storage protocols over the same host adapter as defined in claim 1, wherein said storage related data is transferred using a fibre channel protocol.

20. The system for sending both network and storage data as defined in claim 17, wherein said interface card is configured to be coupled to a fibre channel network.

27. The networked system as defined by claim 25, wherein said communication medium is a fibre channel.

| Full | Title | Citation | Front | Review | Classification | Date | Reference | Sequences | Attachments | Claims | KWIC | Draw De |

☐ 7.   Document ID: US 6470382 B1

L4: Entry 7 of 16   File: USPT   Oct 22, 2002

DOCUMENT-IDENTIFIER: US 6470382 B1
TITLE: Method to dynamically attach, manage, and access a LAN-attached SCSI and netSCSI devices

Brief Summary Text (5):
One method of overcoming the limitations of a SCSI bus is to simply use a network other than a SCSI bus. For example Server area networks (SAN) currently use a Fiber Channel (FC) medium to communicate with SCSI devices. Another type of network that could be used is a Local Area Network (LAN) or an Internet Protocol (IP) Network.

h   e b   b g ee e f   `e   h   ef   b   e

Brief Summary Text (10):
Bus 114 of FIG. 1 could be a non-SCSI network, such as the FC link. An FC network
has features that allow new devices to be recognized and mapped to SCSI operations
automatically. However, this address mapping method is only a rudimentary mapping
of the SCSI ID to the FC address. Additionally, special configuration SCSI devices,
specifically SCSI-3 devices, would be required for automatic addition and removal
capability on network 100. Thus, SCSI devices 102, 104, and 106 would be SCSI-3
configuration. Furthermore, the FC network mapping is not applicable to a LAN or IP
network. Hence, a need arises for a method and apparatus to allow SCSI devices to
be attached to the LAN, be recognized by the hosts/clients and other SCSI devices
on the same LAN or IP network automatically, and for the LAN/IP address of the
netSCSI device to be bound to a logical SCSI ID. An additional need exists to allow
the use of standard SCSI devices on a network such that backwards compatibility is
maintained. If such a network for SCSI devices existed, a further need would exist
for a protocol in which to encapsulate data and SCSI commands)into a coherent
network-compatible packet.

Brief Summary Text (11):
In summary, a need exists for a method and apparatus for accommodating more SCSI
devices on a bus and for placing them at greater distances on the bus than would be
feasible with a conventional SCSI bus. Hence, a need arises for a method and
apparatus for attaching and managing SCSI devices on a network. Hence, a need
arises for a network system and method to automatically and dynamically attach,
manage, and configure a netSCSI device on the network itself. Also, a need exists
for a method and apparatus for dynamically addressing SCSI devices that would
automatically avoid address conflicts when installing new devices and when the bus
is broken. Furthermore, a need arises for a method and apparatus to allow SCSI
devices to be attached to the LAN, be recognized by the hosts/clients and other
SCSI devices on the same LAN or IP network automatically, and for the LAN/IP
address of the netSCSI device to be bound to a logical SCSI ID. An additional need
exists to allow the use of standard SCSI devices on a network such that backwards
compatibility is maintained. Finally, a need exists for a protocol in which to
encapsulate data and SCSI commands into a coherent network-compatible packet. The
present invention provides a unique and novel solution that meets all the above
needs.

Detailed Description Text (80):
In conclusion, the present invention provides a method and apparatus for
accommodating more SCSI devices on a bus and for placing them at greater distances
on the bus than would be feasible on a conventional SCSI bus. The present invention
also provides a method and apparatus for dynamically addressing SCSI devices
automatically to avoid address conflicts. Additionally, the present invention
provides a network system and configuration that can accommodate peripherals
independent from the server that can be placed on the network. Furthermore, the
present invention provides a method and apparatus that allows SCSI devices to be
attached to the LAN, be recognized by the hosts/clients and other SCSI devices on
the same LAN or IP, network automatically, and for the LAN/IP address of the
netSCSI device to be bound to a logical SCSI ID. The present invention also allows
the use of standard SCSI devices on a network such that backwards compatibility is
maintained. Finally, the present invention provides a protocol in which to
encapsulate data and SCSI commands into a coherent network-compatible packet.

| Full | Title | Citation | Front | Review | Classification | Date | Reference | Sequences | Attachments | Claims | KWIC | Draw De |

☐ 8. Document ID: US 6470026 B1

L4: Entry 8 of 16   File: USPT   Oct 22, 2002

h   e b   b   g ee e f   e   h   ef   b   e

DOCUMENT-IDENTIFIER: US 6470026 B1
TITLE: Fibre channel loop map initialization protocol implemented in hardware

Abstract Text (1):
A fiber channel interface controller incorporating a cost-efficient hardware
implementation of the fiber channel arbitrated loop initialization protocol. Cost
efficiency is achieved by determining the minimal buffer size required for storing
information extracted from fiber channel arbitrated loop initialization frames and
including in the fiber channel interface controller a memory buffer having this
calculated minimum size. The inherent buffering capacity within all FC ports of the
arbitrated loop and the ability to generate, on the fly, large portions of fiber
channel arbitrated loop initialization frames, rather than storing and forwarding
entire fiber channel arbitrated loop initialization frames, both combine to
decrease the memory buffer size requirements for the fiber channel interface
controller and to decrease the time required to carry out fiber channel arbitrated
loop initialization. The decreased time required to carry out fiber channel
arbitrated loop initialization results in less down time and increased availability
of the fiber channel that are import.

Brief Summary Text (2):
The present invention relates to fibre channel interface controllers used to
implement fibre channel ports and, in particular, to a method and system for
implementing, in hardware, the final two phases of fibre channel arbitrated loop
initialization protocol that concern the construction and the distribution of an
arbitrated loop physical address position map.

Brief Summary Text (4):
The fibre channel ("FC") is an architecture and protocol for a data communication
network for interconnecting a number of different combinations of computers and
peripheral devices. The FC supports a variety of upper-level protocols, including
the small computer systems interface ("SCSI") protocol. A computer or peripheral
device is linked to the network through an FC port and copper wires or optical
fibres. An FC port includes a transceiver and an interface controller, and the
computer peripheral device in which the FC port is contained is called a "host."
The FC port exchanges data with the host via a local data bus, such as a peripheral
computer interface ("PCI") bus. The interface controller conducts lower-level
protocol exchanges between the fibre channel and the computer or peripheral device
in which the FC port resides.

Brief Summary Text (5):
In one type of FC topology, called the "fibre channel arbitrated loop", the various
interconnected FC ports carry out an arbitrated loop initialization protocol in
order to initialize data communications traffic between the FC ports of the
arbitrated loop. The arbitrated loop initialization protocol involves a number of
different phases. In the first of two final phases of arbitrated loop
initialization, the FC ports construct, one-by-one, a position map that represents
their relative positions within the arbitrated loop, and, then, in the second of
the final two phases, distribute the completed position map amongst themselves. In
previous and current implementations of FC interface controllers, these final two
phases of the arbitrated loop initialization protocol were either not implemented
or were implemented in firmware, so that these final two phases of the arbitrated
loop initialization protocol were either not available or were carried out at
relatively slow speed. For many reasons, including the necessity for high-
availability systems to quickly reinitialize arbitrated loops following a reset of
any FC port within the arbitrated loop, it is desirable for the arbitrated loop
initialization protocol to operate far more quickly than previous and current
implementations allow. A need has therefore been recognized by developers and users
of fibre channel interface controllers for an interface controller that implements,

h   e b   b g ee e f   e   h   ef   b   e

in hardware, the arbitrated loop initialization protocol.

Brief Summary Text (7):
The present invention provides a fibre channel ("FC") interface controller that
implements, in hardware, all phases of the FC arbitrated loop initialization
protocol. In particular, the final two phases of the arbitrated loop initialization
protocol that involve the construction and distribution of a somewhat lengthy
position map are implemented to execute within the FC interface controller, without
host computer or host peripheral device intervention. The efficient and economical
implementation of the present invention takes advantage of the inherent dynamic
buffering capacity of the FC ports interconnected by an FC arbitrated loop and of
the fact that relatively large portions of the data exchanged between FC ports of
the FC arbitrated loop during arbitrated loop initialization need not be buffered
for forwarding, but can instead be generated on the fly during transmission of the
data. The minimization of the amount of memory buffer requirements within the
interface controller contributes significant cost advantages in the design and
production of the interface controller, and simplifies the algorithms and circuitry
that implement the FC arbitrated loop protocol. Because the final two phases of the
arbitrated loop initialization protocol are implemented to execute within the FC
interface controller, without host computer or host peripheral device intervention,
the FC arbitrated loop initialization can proceed much faster, decreasing the time
that the FC is unavailable for data communications. This is, in turn, important in
high-availability systems, where continuous data communications are desirable.

Drawing Description Text (11):
FIG. 10 shows the basic underlying circular queue data structure used in the
Tachyon TL Fibre Channel Mass Storage Interface Controller interface.

Detailed Description Text (2):
The present invention will be described below in six subsections. The first three
subsections provide greater detail about the fibre channel architecture and
protocol, the SCSI architecture and protocol, and implementation of the SCSI
protocol on top of the fibre channel protocol. The fourth subsection discusses the
fibre channel arbitrated loop intialization process. The fifth subsection provides
a general description of the present invention, and the sixth subsection provides a
detailed pseudo-code implementation of the present invention.

Detailed Description Text (3):
Fibre Channel

Detailed Description Text (4):
The Fibre Channel ("FC") is defined by, and described in, a number of ANSI
Standards documents, including: (1) Fibre Channel Physical and Signaling Interface
("FC-PH"), ANSI X3.230-1994, ("FC-PH-2"), ANSI X3.297-1997; (2) Fibre Channel--
Arbitrated Loop ("FC-AL-2"), ANSI X3.272-1996; (3) Fibre Channel--Private Loop SCSI
Direct Attached ("FC-PLDA"); (4) Fibre Channel--Fabric Loop Attachment ("FC-FLA");
(5) Fibre Channel Protocol for SCSI ("FCP"); (6) Fibre Channel Fabric Requirements
("FC-FG"), ANSI X3.289:1996; and (7) Fibre Channel 10-Bit Interface. These
standards documents are under frequent revision. Additional Fibre Channel System
Initiative ("FCSI") standards documents include: (1) Gigabaud Link Module Family
("GLM"), FCSI-301; (2) Common FC-PH Feature Sets Profiles, 101; and (3) SCSI
Profile, FCSI-201. These documents may be found at the world wide web Internet page
having the following address: "http://www.fibrechannel.com"

Detailed Description Text (7):
In the following discussion, "FC" is used as an adjective to refer to the general
Fibre Channel architecture and protocol, and is used as a noun to refer to an
instance of a Fibre Channel communications medium. Thus, an FC (architecture and
protocol) port may receive an FC (architecture and protocol) sequence from the FC
(communications medium).

h   e b   b g ee e f   e   h   ef   b   e

Detailed Description Text (14):
The FC is a full duplex data transmission medium. Frames and sequences can be
simultaneously passed in both directions between an originator, or initiator, and a
responder, or target. An exchange comprises all sequences, and frames within the
sequences, exchanged between an originator and a responder during a single I/O
transaction, such as a read I/O transaction or a write I/O transaction. The FC
protocol is designed to transfer data according to any number of higher-level data
exchange protocols, including the Internet protocol ("IP"), the Small Computer
Systems Interface ("SCSI") protocol, the High Performance Parallel Interface
("HIPPI"), and the Intelligent Peripheral Interface ("IPI"). The SCSI bus
architecture will be discussed in the following subsection, and much of the
subsequent discussion in this and remaining subsections will focus on the SCSI
protocol embedded within the FC protocol. The standard adaptation of SCSI protocol
to fibre channel is subsequently referred to in this document as "FCP." Thus, the
FC can support a master-slave type communications paradigm that is characteristic
of the SCSI bus and other peripheral interconnection buses, as well as the
relatively open and unstructured communication protocols such as those used to
implement the Internet. The SCSI bus architecture concepts of an initiator and
target are carried forward in the FCP, designed, as noted above, to encapsulate
SCSI commands and data exchanges for transport through the FC.

Detailed Description Text (16):
The next high level section 308, called the data payload, contains the actual data
packaged within the FC frame. The data payload contains data and encapsulating
protocol information that is being transferred according to a higher-level
protocol, such as IP and SCSI. FIG. 3 shows four basic types of data payload
layouts 326-329 used for data transfer according to the SCSI protocol. The first of
these formats 326, called the FCP_CMND, is used to send a SCSI command from an
initiator to a target. The FCP_LUN field 330 comprises an 8-byte address that may,
in certain implementations, specify a particular SCSI-bus adapter, a target device
associated with that SCSI-bus adapter, and a logical unit number ("LUN")
corresponding to a logical device associated with the specified target SCSI device
that together represent the target for the FCP_CMND. In other implementations, the
FCP_LUN field 330 contains an index or reference number that can be used by the
target FC host adapter to determine the SCSI-bus adapter, a target device
associated with that SCSI-bus adapter, and a LUN corresponding to a logical device
associated with the specified target SCSI device. An actual SCSI command, such as a
SCSI read or write I/O command, is contained within the 16-byte field FCP_CDB 332.

Detailed Description Text (41):
The Tachyon TL Mass Storage Interface Controller ("TL") is a high-performance, low-
cost, loop-based interface controller for use in the FC ports that interconnect
peripheral devices and computers to an FC arbitrated loop. In this subsection, an
overview of the functionality of, and interface to, the TL will be presented. A
more detailed description of the TL is included in U.S. Pat. application Ser. No.
68/582,001, entitled "Fibre Channel Controller," assigned to the Hewlett Packard
Company, and filed on Oct. 30, 1998 that is hereby incorporated by reference in its
entirety.

Detailed Description Text (46):
The completion message manager 925 manages the IMQ and provides queue entries to
the inbound data manager 924 into which the inbound data manager places completion
messages. The single frame manager 926 manages the SFQ in host memory and provides
entries to the fibre channel services component 927 into which the fibre channel
component services place inbound frames. The exchange request manager 931 fetches
new entries from the ERQ and sends them to the SCSI exchange manger-outbound ("SEM-
OUT") for processing. The inbound data manager 924 informs the inbound frame
processors, i.e. the SCSI exchange manager-inbound ("SEM-IN") 928 and fibre channel
services component 927, of new frames and routes the frames to their proper

h   e b   b g ee e f   e   h   ef   b   e

destination in the host. Also, the inbound data manager sends completion messages
to the host via the IMQ. The <u>fibre channel</u> services component 927 manages the <u>fibre
channel</u> frames that the SEM-IN 928 does not manage. The <u>fibre channel</u> services
component places the frames in the SFQ. The SEM-IN 928 manages the phases of a SCSI
exchange that receive a <u>fibre channel</u> sequence. The SEM-IN reads the SEST entries
via the SEST link fetch manager 929 and either sends the inbound data to the proper
host buffers or sends the request to the SEM-OUT 932 to send the next phases of
<u>fibre channel</u> sequence. The SEST link fetch manager 929 is responsible for reading
and writing SEST entries, depending upon requests from the SEM-IN 928 and SEM-OUT
932 components. The SEM-OUT 932 manages the phases of a SCSI exchange that require
a <u>fibre channel</u> sequence to be sent. The SEM-OUT 932 reads the SEST entries via the
SEST link fetch manager 929, builds the request to send those sequences, and sends
the requests to the outbound sequence manager 930. The outbound sequence manager
("OSM") 930 processes requests from the SEM-OUT 932 to send <u>fibre channel</u> sequences
from the host and retrieves <u>fibre channel</u> frame headers and payloads from the host
to send to the remote node. The OSM segments the sequence into <u>fibre channel</u> frames
of up to 1 KByte in size and queues them into the outbound FIFO manager 916.

<u>Detailed Description Text</u> (49):
FIG. 11 shows a more detailed view of the host memory data structures required to
perform an FCP write operation where the FC node in which a TL resides is the
initiator of the FCP write operation and where the data payload that will include
the data to be written requires 4 or more data buffers. The host prepares an
initiator write entry ("IWE") 1102 within a SEST entry 1104 in the SEST (904 in
FIG. 9). Associated with the IWE are: (1) a <u>fibre channel</u> header structure ("FCHS")
1106 that is used to send the FCP_DATA sequence; (2) a data buffer 1108 that is
used to receive the FCP_RSP frame from the SCSI target; and (3) one or more
extended scatter gather list ("SGL") pages 1110 and 1112 that contain pointers to
data buffers 1113-1117 in which the host places the data to be written to the SCSI
target via the FCP_DATA sequence. The host then creates an I/O request block
("IRB") 1118 in an unused ERQ entry 1120 and associates the IRB with an FCHS 1122
that is used for the FCP_CMND sequence. The host then increments the ERQ producer
index. The producer index increment is detected by the TL, and the TL then launches
the FCP write operation. The TL uses the information and data stored within the IRB
1118 and RWE 1102, and the data structures associated with the IRB and RWE, to
conduct the entire FCP write operation, including the FCP_CMND sequence, the
FCP_XFER_RDY sequence, and the FCP_DATA sequence. The TL receives from the target a
FCP_RSP sequence at the completion of the FCP write operation.

<u>Detailed Description Text</u> (65):
Hardware Implementation of the <u>Fibre Channel</u> Loop Map Initialization Protocol

<u>Detailed Description Text</u> (84):
The class "byte array," declared above on lines 1-12, implements fixed size byte
arrays. Member functions declared for the class "byte_array" include:
(1)"byte_array," a constructor that takes, as a single argument, that number of
bytes that the fixed-size byte array will include; (2) an array indexing operator
for accessing individual bytes within the byte_array just as elements of a
language-defined array are accessed with the array indexing operator; (3) an
equivalence conditional operator that compares one byte_array to another and
returns the Boolean value TRUE if the two byte_arrays contain numerically
equivalent values; (4) a "less than" conditional operator that compares the instant
byte_array to a second byte_array and returns the Boolean value TRUE if the
numerical value represented by the second byte_array is greater than that of the
instant byte_array; (5) "setHighByte," a member function that sets the most
significant byte of a byte_array to the value supplied in argument "x"; (6)
"setLowByte," a member function that sets the least significant byte of a
byte_array to the value supplied in argument "x"; (7) "setBit," a member function
that sets the bit of the instant byte array specified by the argument "bit" to "1";
(8) "bitSet," a member function that returns the Boolean value TRUE is the bit of

h   e b   B g e e e f   e   h   ef   b   e

the instant byte array specified by the argument "bit" has the value "1"; and (9) "bitClear," a member function that sets the bit of the instant byte array specified by the argument "bit" to "0." The class "queue" declared above on lines 14-20, represents a generalized input queue from which the loop initialization state machine, to be described below, receives incoming bytes from the fibre channel. The class "queue" can be considered to be a generalization of the elastic store (934 in FIG. 9) and the minimal buffering component (2506 in FIG. 25). The elastic store (934 in FIG. 9) is a small buffer for buffering received bytes from the fibre channel that cannot be immediately forwarded by the transceiver (806 in FIG. 8) due to a lack of synchronization between the clocks of the receiver and transmitter components of the transceiver (806 in FIG. 8). The elastic store is a major source of inherent latency described in the previous subsection of each FC port. As also described in the previous subsection, minimal additional buffering is required to receive a frame from the FC port while transmission of an FC arbitrated loop initialization frame by a given FC port is carried out. Thus, the class "queue" represents a buffer, or byte sink, that comprises both the elastic store (934 in FIG. 9) and the minimal memory buffer component (2306 in FIG. 25). Four methods are declared for the class "queue:" (1) "more," a method that returns a Boolean value indicating whether there are additional bytes in the queue available for processing; (2) "ARB_F0," a method that returns a Boolean value indicating whether an ARB_OF FC primitive has been received by the receiver and placed, in order of reception, into the queue; (3) "get next," a method that returns the next available byte within the queue; and (4) "CLS_Pr," a method that returns a Boolean value indicating whether a CLS FC primitive has been received by the receiver and placed, in order of reception, into the queue. A primitive, such as the ARB_F0 and CLS primitives, is a set of four bytes that do not belong to an FC frame and that include a type indication for distinguishing different types of primitives.

Detailed Description Text (87):
The class "transmitter," declared above on lines 32-38, includes the methods: (1) "transmit_byte," a method that transmits a single byte "b" to the FC; (2) "end frame," a method that indicates to the transmitter that the FC arbitrated loop initialization frame currently being transmitted is complete, and that the transmitter should transmit the CRC and EOF fields (310 and 312 in FIG. 3) to complete transmission of the frame; (3) "transmit_ARB_F0," a method that transmits an ARB_F0 primitive to the FC; and (4) "transmit_CLS," a method that transmits a CLS primitive to the FC. Finally, the class "receiver," declared above on lines 40-43, includes the single method "signal receiver" that indicates to the receiver component that the FC port has completed the fibre channel arbitrated loop initialization protocol and indicates to the receiver by the argument "participating" whether or not the FC port has acquired an AL_PA during the initialization process and will be participating in the FC arbitrated loop.

Detailed Description Text (110):
If the byte in the queue "q" indicates that the receiver has received an ARB_F0 primitive, as detected by producer_signal on line 10, producer_signal calls the getNext function of the queue "q" on line 12 to remove the ARB_F0 primitive. If the FC port is in the ARBF0 loop initialization state and the fibre channel state is NO_FRAME, as detected on line 13-14, then the ARB_F0 primitive is handled by producer_signal on lines 16-23. Otherwise, this ARBF0 primitive was sent in error, and producer_signal calls the error member function on line 25. If the FC port is not the LIM, as detected on line 16, then the FC port forwards the ARB_F0 primitive on line 18, disables and resets the timer on lines 19 and 20, and continues a subsequent iteration of the encompassing while-loop on line 8 via a continue statement on line 21. If the FC port is the LIM, then producer_signal sets the loop initialization phase for the FC port to LIFA on line 23, since receipt of the ARB_F0 primitive means that the ARB_P0 primitive has traveled all the way around the FC arbitrated loop.

CLAIMS:

h   e b   b g ee e f   e   h   ef   b   e

1. An interface controller that operates within a <u>fibre channel</u> node to implement all phases of the <u>fibre channel</u> loop initialization protocol, the interface controller coupled to receiver and transmitter components that are both, in turn, coupled to a communications network transport medium through which the <u>fibre channel</u> node exchanges frames composed of an ordered set of data units with other <u>fibre channel</u> nodes, the interface controller comprising: a frame manager that receives data units from the receiver and that transmits data units to a transmitter, the frame manager, receiver, and transmitter together providing an inherent data buffering capacity to the <u>fibre channel</u> node; a logic component that processes data units received by the frame manager, that initiates transmission of data units by the frame manager, and that coordinates the transmission of data units with respect to the reception of data units in order to implement all phases of the <u>fibre channel</u> loop initialization protocol; a memory buffer having a size of at least 13 bytes that, combined with the inherent data buffering capacity of the <u>fibre channel</u> nodes coupled to the transport medium of the communications network, is sufficient in size to hold data units of a frame being received by the frame manager until the data units are processed by the logic component; and a generator that generates and transmits to the transmitter certain data units required by the <u>fibre channel</u> loop initialization protocol for inclusion in a frame being transmitted by the frame manager.

3. The interface controller of claim 2 wherein the interface controller operates within a <u>fibre channel</u> communications network.

4. The interface controller of claim 3 wherein the interface controller operates within a <u>fibre channel</u> arbitrated loop.

5. The interface controller of claim 4 wherein the generator generates frame headers, loop initialization identifier fields, and loop initialization flags fields of all <u>fibre channel</u> arbitrated loop initialization frames, as well as port name fields of the loop initialization select master initialization frames and empty arbitrated loop physical address position map slots within loop initialization report position and loop initialization loop position <u>fibre channel</u> arbitrated loop initialization frames.

7. A method for processing frames within an interface controller according to all phases of <u>fibre channel</u> loop initialization protocol, the interface controller used in a <u>fibre channel</u> port within a <u>fibre channel</u> node to exchange frames composed of an ordered set of data units with other <u>fibre channel</u> nodes, the method comprising: coupling the interface controller to a receiver and a transmitter that are both, in turn, coupled to a communications network transport medium through which the <u>fibre channel</u> node exchanges frames units with the other <u>fibre channel</u> nodes; receiving data units of a frame from the receiver; when the data units are immediately processed, processing certain of the received data units according to the <u>fibre channel</u> loop initialization protocol; when the received data units are not immediately processed, because another task is simultaneously being carried out by the interface controller, storing in a memory buffer only the received data units that represent values required for processing the frame according to <u>fibre channel</u> loop initialization protocol or that cannot be generated by the hardware logic circuit based on inherent and acquired characteristics of the <u>fibre channel</u> loop initialization port, and later processing certain of the received data units according to the <u>fibre channel</u> loop initialization protocol; and when the received frame needs to be forwarded to another <u>fibre channel</u> node, according to the <u>fibre channel</u> loop initialization protocol, transmitting the received data units that cannot be generated by a hardware logic circuit based on inherent and acquired characteristics of the <u>fibre channel</u> port to the transmitter for inclusion in the frame to be forwarded and generating and transmitting to the transmitter by a hardware logic circuit the remaining data units of the frame to be forded.

h   e b   b g ee e f   e   h   ef   b   e

8. The method of claim 7 wherein the interface controller operates within a <u>fibre channel</u> communications network.

9. The method of claim 8 wherein the interface controller operates within a <u>fibre channel</u> arbitrated loop.

10. The method of claim 9 further including: generating frame headers, loop initialization identifier fields, and loop initialization flags fields of all <u>fibre channel</u> arbitrated loop initialization frames, as well as port name fields of loop initialization select master initialization frames and empty arbitrated loop physical address position map slots within loop initialization report position and loop initialization loop position <u>fibre channel</u> arbitrated loop initialization frames.

13. A method for decreasing loop initialization latency and for minimizing a size of a memory buffer required within a component of a <u>fibre channel</u> arbitrated loop interface controller that implements the loop initialization protocol within a <u>fibre channel</u> port and that contains a frame manager coupled to receiver and transmitter components, the method comprising: providing within the-interface controller state machines and circuitry for carrying out all phases of loop initialization without the need for external intervention by a host processor; and generating, rather than storing and forwarding, those portions of loop initialization frames that can be generated based on acquired and inherent characteristics of the <u>fibre channel</u> port that contains the interface controller.

14. The method of claim 13 wherein generating, rather than storing and forwarding, those portions of loop initialization frames that can be generated based on acquired and inherent characteristics of the <u>fibre channel</u> port includes generating: frame headers of all <u>fibre channel</u> arbitrated loop initialization frames, loop initialization identifier fields of all <u>fibre channel</u> arbitrated loop initialization frames, loop initialization flags fields of all <u>fibre channel</u> arbitrated loop initialization frames, port name fields of the loop initialization select master initialization frames, and empty arbitrated loop physical address position map slots within loop initialization report position and loop initialization loop position <u>fibre channel</u> arbitrated loop initialization frames.

15. The method of claim 13 further including: subtracting a portion of the inherent buffering capacity within the frame manager, receiver, and transmitter components of all <u>fibre channel</u> ports receiving and transmitting <u>fibre channel</u> arbitrated loop initialization frames from the memory buffering capacity required by the component of the <u>fibre channel</u> arbitrated loop interface controller that implements the loop initialization protocol to calculate a minimum buffering capacity required and providing in the component of the <u>fibre channel</u> arbitrated loop interface controller that implements the loop initialization protocol a memory buffer having the calculated minimum buffering capacity.

| Full | Title | Citation | Front | Review | Classification | Date | Reference | Sequences | Attachments | Claims | KWIC | Draw De |

☐ 9. Document ID: US 6463498 B1

L4: Entry 9 of 16   File: USPT   Oct 8, 2002

DOCUMENT-IDENTIFIER: US 6463498 B1
TITLE: Transmission of FCP response in the same loop tenancy as the FCP data with minimization of inter-sequence gap

h   e b   b g e e f   e   h   ef   b   e

Brief Summary Text (4):
The Fibre Channel ("FC") is an architecture and protocol for a data communication
network for interconnecting a number of different combinations of computers and
peripheral devices. The FC supports a variety of upper-level protocols, including
the small computer systems interface ("SCSI") protocol. A computer or peripheral
device is linked to the network through an FC Port and copper wires or optical
fibres. An FC Port includes a transceiver and an interface controller, and the
computer peripheral device in which the FC Port is contained is called a "host."
The FC Port exchanges data with the host via a local data bus, such as a peripheral
computer interface ("PCI") bus. The interface controller conducts lower-level
protocol exchanges between the Fibre Channel and the computer or peripheral device
in which the FC Port resides.

Brief Summary Text (5):
The FC is becoming increasingly important as a means to interconnect computers and
peripheral storage devices. In such applications, an FC arbitrated loop topology is
commonly employed. In an FC arbitrated loop, an FC node contends for control of the
FC arbitrated loop and, upon gaining control, opens a channel to another FC node
and exchanges data with that FC node in half-duplex or full-duplex mode. Peripheral
mass storage devices commonly receive and respond to write and read commands from a
host computer according to the SCSI protocol. Thus, in the common FC arbitrated
loop topology for interconnecting host computers with peripheral mass storage
devices, the host computer and peripheral mass storage devices exchange read and
write commands and data through the SCSI protocol implemented on top of the Fibre
Channel protocol.

Brief Summary Text (8):
The present invention provides a Fibre Channel ("FC") interface controller for use
in FC Ports interconnected by an FC arbitrated loop, that implements, in hardware,
an efficient method for executing read commands in order to avoid unnecessary Fibre
Channel throughput degradation and unnecessary increase in I/O latencies. In the
improved method, the FC host containing an FC Port implemented with the improved
interface controller of the present invention prepares a description of the data
requested by a received read operation command into host memory buffers and
assembles an FCP_Response sequence at the same time. The FC host then provides the
improved interface controller with a data structure that includes references to the
host memory buffers that contain the data requested by the read operation command
and that also includes a reference to the description of the FCP_Response sequence.
The FC Port implemented using the improved interface controller, upon receiving the
data structure from the host, arbitrates for control of the FC arbitrated loop,
returns the data requested by the read operation command to the initiator of the
read operation command in an FC data sequence, and then immediately transmits to
the initiator the already prepared FCP_Response sequence. By this method, the FC
Port including the improved interface controller avoids arbitrating twice for
control of the FC arbitrated loop in order to respond to the received read
operation command and also avoids the increased I/O latency incurred by certain
currently-available interface controllers when they return control to the host for
preparation of the FCP_Response sequence. The improved interface controller
transfers both the FCP_data sequence and the FCP_Response sequence directly from
host memory buffers to the Fibre Channel arbitrated loop, rather than unnecessarily
storing and forwarding the FCP data sequence and FCP_Response sequence.

Drawing Description Text (11):
FIG. 10 shows the basic underlying circular queue data structure used in the
Tachyon TL Fibre Channel Mass Storage Interface Controller interface.

Detailed Description Text (2):
The present invention will be described below in six subsections. The first three
subsections provide greater detail about the Fibre Channel architecture and

h   e b   b g ee e f   e   h   ef   b   e

protocol, the SCSI architecture, and implementation of the SCSI protocol on top of the Fibre Channel protocol. The fourth subsection discusses the Fibre Channel arbitrated loop intialization process. The fifth subsection provides a general description of the present invention, and the sixth subsection provides an implementation of the present invention.

Detailed Description Text (3):
Fibre Channel

Detailed Description Text (4):
The Fibre Channel ("FC") is defined by, and described in, a number of ANSI Standards documents, including: (1). Fibre Channel Physical and Signaling Interface ("FC-PH"), ANSI X3.230-1994, ("FC-PH-2"), ANSI X3.297-1997; (2) Fibre Channel-- Arbitrated Loop ("FC-AL-2"), ANSI X3.272-1996; (3) Fibre Channel Private Loop SCSI Direct Attached ("FC-PLDA"); (4) Fibre Channel--Fabric Loop Attachment ("FC-FLA"); (5) Fibre Channel Protocol for SCSI ("FCP"); (6) Fibre Channel Fabric Requirements ("FC-FG"), ANSI X3.289:1996; and (7) Fibre Channel, 10-Bit Interface. These standards documents are under frequent revision. Additional Fibre Channel System Initiative ("FCSI") standards documents include: (1) Gigabaud Link Module Family ("GLM"), FCSI-301; (2) Common FC-PH Feature Sets Profiles, FCSI-101; and (3) SCSI Profile, FCSI-201. FC standards documents may be found at the world wide web Internet page having the following address:

Detailed Description Text (8):
In the following discussion, "FC" is used as an adjective to refer to the general Fibre Channel architecture and protocol, and is used as a noun to refer to an instance of a Fibre Channel communications medium. Thus, an FC (architecture and protocol) port may receive an FC (architecture and protocol) sequence from the FC (communications medium).

Detailed Description Text (14):
The FC is a full duplex data transmission medium. Frames and sequences can be simultaneously passed in both directions between an originator, or initiator, and a responder, or target. An exchange comprises all sequences, and frames within the sequences, exchanged, between an originator and a responder during a single I/O transaction, such as a read I/O transaction or a write I/O transaction. The FC protocol is designed to transfer data according to any number of higher-level data exchange protocols, including the Internet protocol ("IP"), the Small Computer Systems Interface ("SCSI") protocol, the High Performance Parallel Interface ("HIPPI"), and the Intelligent Peripheral Interface ("IPI"). The SCSI bus architecture will be discussed in the following subsection, and much of the subsequent discussion in this and remaining subsections will focus on the SCSI protocol embedded within the FC protocol. The standard adaptation of SCSI protocol to Fibre Channel is subsequently referred to in this document as "FCP." Thus, the FC can support a master-slave type communications paradigm that is characteristic of the SCSI bus and other peripheral interconnection buses, as well as the relatively open and unstructured communication protocols such as those used to implement the Internet. The SCSI bus architecture concepts of an initiator and target are carried forward in the FCP, designed, as noted above, to encapsulate SCSI commands and data exchanges for transport through the FC. FIG. 3 shows the contents of a standard FC frame. The FC frame 302 comprises five high level sections 304, 306, 308, 310 and 312. The first high level section, called the start-of-frame deliminator 304, comprises 4 bytes that mark the beginning of the frame. The next high level section, called frame header 306, comprises 24 bytes that contain addressing information, sequence information, exchange information, and various control flags. A more detailed view of the frame header 314 is shown expanded from the FC frame 302 in FIG. 3. The destination identifier ("D_ID"), or DESTINATION_ID 316, is a 24-bit FC address indicating the destination FC Port for the frame. The source identifier ("S_ID"), or SOURCE_ID 318, is a 24-bit address that indicates the FC Port that transmitted the frame. The originator ID, or OX_ID

h   e b   b g ee e f   e   h   ef   b   e

320, and the responder ID 322, or RX_ID, together compose a 32-bit exchange ID that
identifies the exchange to which the frame belongs with respect to the originator,
or initiator, and responder, or target, FC Ports. The sequence ID, or SEQ_ID, 324
identifies the sequence to which the frame belongs.

Detailed Description Text (15):
The next high level section 308, called the data payload, contains the actual data
packaged within the FC frame. The data payload contains data and encapsulating
protocol information that is being transferred according to a higher-level
protocol, such as IP and SCSI. FIG. 3 shows four basic types of data payload
layouts 326-329 used for data transfer according to the SCSI protocol. The first of
these formats 326, called the FCP_CMND, is used to send a SCSI command from an
initiator to a target. The FCP_LUN field 330 comprises an 8-byte address that may,
in certain implementations, specify a particular SCSI-bus adapter, a target device
associated with that SCSI-bus adapter., and a logical unit number ("LUN")
corresponding to a logical device associated with the specified target SCSI device
that, together represent the target for the FCP_CMND. In other implementations, the
FCP_LUN field 330 contains an index or reference number that can be used by the
target FC host adapter to determine the SCSI-bus adapter, a target device
associated with that SCSI-bus adapter, and a LUN corresponding to a logical device
associated with the specified target SCSI device. An actual SCSI command, such as a
SCSI read or write I/O command, is contained within the 16-byte field FCP_CDB 332.

Detailed Description Text (40):
The Tachyon TL FC Mass Storage Interface Controller ("TL") is a high-performance,
low-cost, loop-based interface controller for use in the FC Ports that interconnect
peripheral devices and computers to an FC arbitrated loop. In this subsection, an
overview of the functionality of, and interface to, the TL will be presented. A
more detailed description of the TL is included in U.S. patent application Ser. No.
09/183,164, entitled "Fibre Channel Controller," assigned to the Hewlett Packard
Company, and filed on Oct. 30, 1998 that is hereby incorporated by reference in its
entirety.

Detailed Description Text (45):
The completion message manager 925 manages the IMQ and provides queue entries to
the inbound data manager 924 into which the inbound data manager places completion
messages. The single frame manager 926 manages the SFQ in host memory and provides
entries to the Fibre Channel services component 927 into which the Fibre Channel
component services place inbound frames. The exchange request manager 931 fetches
new entries from the ERQ and sends them to the SCSI exchange manger-outbound ("SEM-
OUT") for processing. The inbound data manager 924 informs the inbound frame
processors, i.e. the SCSI exchange manager-inbound ("SEM-IN") 928 and Fibre Channel
services component 927, of new frames and routes the frames to their proper
destination in the host. Also, the inbound data manager sends completion messages
to the host via the IMQ. The Fibre Channel services component 927 manages the Fibre
Channel frames that the SEM-IN 928 does not manage. The Fibre Channel services
component places the frames in the SFQ. The SEM-IN 928 manages the phases of a SCSI
exchange that receive a Fibre Channel sequence. The SEM-IN reads the SEST entries
via the SEST link fetch manager 929 and either sends the inbound data to the proper
host buffers or sends the request to the SEM-OUT 932 to send the next phases of
Fibre Channel sequence. The SEST link fetch manager 929 is responsible for reading
and writing SEST entries, depending upon requests from the SEM-IN 928 and SEM-OUT
932 components. The SEM-OUT 932 manages the phases of a SCSI exchange that require
a Fibre Channel sequence to be sent. The SEM-OUT 932 reads the SEST entries via the
SEST link fetch manager 929, builds the request to send those sequences, and sends
the requests to the outbound sequence manager 930. The outbound sequence manager
("OSM") 930 processes requests from the SEM-OUT 932 to send Fibre Channel sequences
from the host and retrieves Fibre Channel frame headers and payloads from the host
to send to the remote node. The OSM segments the sequence into Fibre Channel frames
of up to 1 KByte in size and queues them into the outbound FIFO manager 916.

h   e b   b g ee e f   e   h   ef   b   e

Detailed Description Text (48):
FIG. 11 shows a more detailed view of the host memory data structures required to
perform an FCP write operation where the FC node in which a TL resides is the
initiator of the FCP write operation and where the data payload that will include
the data to be written requires 4 or more data buffers. The host prepares an
initiator write entry ("IWE") 1102 within a SEST entry 1104 in the SEST (904 in
FIG. 9). Associated with the IWE are: (1) a Fibre Channel header structure ("FCHS")
1106 that is used to send the FCP_DATA sequence; (2) a data buffer 1108 that is
used to receive the FCP_RSP frame from the SCSI target; and (3) one or more
extended scatter gather list ("SGL") pages 1110 and 1112 that contain pointers to
data buffers 1113-1117 in which the host places the data to be written to the SCSI
target via the FCP_DATA sequence. The host then creates an I/O request block
("IRB") 1118 in an unused ERQ entry 1120 and associates the IRB with an FCHS 1122
that is used for the FCP_CMND sequence. The host then increments the ERQ producer
index. The producer index increment is detected by the TL, and the TL then launches
the FCP write operation. The TL uses the information and data stored within the IRB
1118 and RWE 1102, and the data structures associated with the IRB and RWE, to
conduct the entire FCP write operation, including the FCP_CMND sequence, the
FCP_XFER_RDY sequence, and the FCP_DATA sequence. The TL receives from the target a
FCP_RSP sequence at the completion of the FCP write operation.

Detailed Description Text (65):
FIG. 21 illustrates the three phases of SCSI read command execution for an
initiator and target interconnected by an FC arbitrated loop topology. In FIG. 21,
an initiator FC node 2102 is interconnected with a target FC node 2104 by an FC
arbitrated loop 2106. The FC arbitrated loop 2106 includes six additional FC-Nodes
2108-2113. In FIG. 21, FC arbitrated loop primitives and Fibre Channel sequences
that are sent from one FC node to another FC node during execution of a SCSI read
operation are shown as arrows, such as arrow 2114. In FIG. 21, time is a radial
axis 2116 directed toward the center of the FIG. 2118. In other words, execution of
the SCSI read operation starts towards the outer circumference of the circular
portion of the figure and proceeds towards the center of the FIG. 2118. FIGS. 22
and 23 employ the same method of illustration as FIG. 21, and elements of FIG. 22
and 23 that are common with elements of FIG. 21 will not be re-labeled in FIGS. 22-
23. FIG. 21 illustrates the execution of an SCSI read operation by FC nodes
employing one type of currently available FC Port. First, the initiator FC node
2102 sends an FC arbitrated loop arbitration primitive 2120 in order to gain
control of the arbitrated loop. Assuming that the FC arbitrated loop 2106 is
quiescent and that the initiator now gains control of the loop as quickly as
possible, arbitration for control of the FC arbitrated loop by the initiator 2102
involves the receiving and forwarding of the FC arbitrated loop primitive 2120 by
each node in the FC arbitrated loop, represented in FIG. 21 by arrows 2122-2124,
2114, and 2125-2127. Next, the initiator FC node 2102 sends an FC arbitrated loop
open primitive 2128 to the target FC node 2104 in order to open a channel between
the initiator FC node 2102 and the FC target node 2104. Next, the initiator FC node
2102 sends an FCP_CMND sequence 2130 to the FC target node 2104 in order to
initiate a read operation by the target FC node 2104. Finally, the initiator FC
node 2102 sends an FC arbitrated loop primitive 2132 to close the channel.

CLAIMS:

3. The interface controller of claim 2 used to implement a Fibre Channel port,
wherein the data sequence is a Fibre Channel sequence composed of a number of Fibre
Channel data frames.

4. The interface controller of claim 3 wherein the read sequence is a Fibre Channel
protocol command sequence, wherein the data sequence is a Fibre Channel protocol
data sequence, and wherein the response sequence is a Fibre Channel protocol
response sequence.

h   e b   b g ee e f   e   h   ef   b   e

8. The method of claim 7 wherein the network communications protocol is the <u>Fibre Channel</u> protocol and wherein the interface controller is used to implement a <u>Fibre Channel</u> network communications port.

9. The method of claim 8 wherein the initiator node and the target node are both interconnected via a <u>Fibre Channel</u> arbitrated loop.

| Full | Title | Citation | Front | Review | Classification | Date | Reference | Sequences | Attachments | Claims | KWIC | Draw De |

☐ 10. Document ID: US 6389432 B1

L4: Entry 10 of 16   File: USPT   May 14, 2002

DOCUMENT-IDENTIFIER: US 6389432 B1
** See image for <u>Certificate of Correction</u> **
TITLE: Intelligent virtual volume access

<u>Brief Summary Text</u> (6):
In a SAN environment, the function of storage is detached from network servers and centralized and managed as a separate network resource. SANs that are based on <u>Fibre Channel</u> buses have recently emerged as one high performance data communications environment available today to interconnect servers and storage. Running at Gigabit speeds and built on open standards, SANs offer better scalability, fault recovery and general manageability than conventional client-server LAN based approaches.

<u>Brief Summary Text</u> (11):
Implementations of the invention may include one or more of the following. The searching step can be performed by an external manager. The attributes of a device can be statically configured in a server. The attributes of a device can be statically configured in a switch. Each data storage device can broadcast its attributes to a manager. The manager can store the attributes in the table, or can statically configures the attributes of a device. Each data storage device can broadcast value-added services associated with the data storage device to the manager. The manager can communicate using a <u>Fibre Channel</u> protocol. Further, each manager can manage one or more virtual volumes on each data storage device. The requester and the data storage devices can communicate using a network protocol. The network protocol can be an Internet Protocol (IP), an Asynchronous Transfer Mode (ATM) protocol, or an in-band control protocol which may be a Small Computer System Interface (SCSI) protocol. A server can manage the data storage device and can communicate using the <u>Fibre Channel</u> protocol. Alternatively, a switch can manages the data storage devices. The switch can communicate using the <u>Fibre Channel</u> protocol. The manager can communicate with other managers over a network. In this case, the searching step includes querying one or more tables in one or more remote storage area networks over the network; returning a remote identification associated with a portion of the selected data storage devices to the requester; and accessing the portion of the selected data storage device using the remote identification. The data storage devices may be a disk drive or a tape drive. The identification can a logical unit number. The requester can be a server or another manager. The method also includes performing manager operations to allocate space. The manager operations include one operation selected from a group of commands to initialize, attach, detach, connect, bind, write, read, move,

h   e b   b g ee e f   e   h   ef   b   e

signal, query, convert, control, and callback data in a storage partition.

Brief Summary Text (13):
Implementations of the invention may include one or more of the following. One or more SANs can be interconnected over a network using a router. The manager can conform to a virtual volume management protocol. The manager can control one or more virtual partitions. Further, the manager can reside on a switch, which may be a Fibre Channel switch or a network switch. One or more of the data storage devices can be coupled to the requester over the SAN or connected over a local area network, a wide area network or a storage framework such as StoreX. The manager can allocate storage space using one or more commands selected from a group of commands to initialize, attach, detach, connect, bind, write, read, move, signal, query, convert, control, and callback data in a storage partition.

Drawing Description Text (5):
FIG. 3 is a diagram illustrating protocol layers in Fibre Channel communications.

Detailed Description Text (4):
Referring now to FIG. 2, the SAN 100, which is similar to the SANs 110 and 120, is shown in more detail. In FIG. 2, the SAN 100 communicates over a bus called Fibre Channel using a Fibre Channel switch 200. The Fibre Channel provides a common transport vehicle for existing channel protocols like Intelligent Peripheral Interface (IPI), Small Computer System Interface (SCSI), and High-Performance Parallel Interface (HIPPI), while offering the characteristics of a framing type protocol.

Detailed Description Text (5):
The Fibre Channel switch 200 may be connected to one or clients 202, which may be applications running on workstations. The Fibre Channel switch 200 may also be connected to one or more near-online devices such as a tape farm 204. Other near-online devices cover optical jukeboxes and robotic tape silos. These devices offer very large capacities which are typically measured in tens or hundred of gigabytes, or even terabytes. However, data access times are slower. Offline storage devices typically consist of tape and optical disk devices used for backup and destined for long-term off-site data storage archiving. A large variety of devices are available for offline storage. While these devices provide the lowest dollar per megabyte cost of capacity in the data storage hierarchy, retrieval of needed data is typically difficult and time-consuming.

Detailed Description Text (6):
To communicate with other LANs and WANs, the Fibre Channel switch 200 is also connected to a router 206. The router 206 can be a special-purpose computer (or software package) that handles the connection between 2 or more networks. The router 206 examines the destination addresses of the packets passing through it and decides which route in the network to send them on.

Detailed Description Text (7):
Additionally, one or more data storage devices such as a disk farm 208 may be connected to the Fibre Channel switch 200. A number of different data storage devices may be attached to the SAN 100. For instance, on-line devices such as magnetic and optical disks may be used. The most popular configuration of on-line devices today is the RAID (Redundant Array of Independent Disks) disk array which can provide enhanced performance, fault tolerance, efficiency and reliability, among others. For example, concurrent access of multiple drives reduces the average rotation time (latency) compared to a single drive. As an example, writing a block of data to N different areas on one large disk requires N spindle rotations to store the data. The same task can be completed on N disks in an array with one rotation. In all RAID installations, parity check data is also stored (either striped across the disks or on a dedicated disk in the array). This check data can be used to rebuild "lost" data in the event of a failed disk drive. Further fault

h   e b   b g ee e f   e   h   ef   b   e

tolerance can be achieved through the "hot swap" replacement of a failed disk with a new disk operation without powering down the RAID array.

Detailed Description Text (8):
One or more point-to-point connections 210, which communicate with a remote node 212 may be connected to the Fibre Channel switch 200. Additionally, a server farm 214 may be connected to the Fibre Channel switch 200. The server farm 214 may be one or more computers or software packages that provide various services to client software running on other computers. The farm 214 may include software, such as a WWW servers, or machines on which the software is running. Moreover, a single server machine could have several different server software packages running on it, thus providing many different servers to clients on the network.

Detailed Description Text (9):
Turning now to FIG. 3, a diagram illustrating protocol layers in Fibre Channel communications is shown. A Fibre Channel bus operates at a wide variety of speeds (133 Mbit/s, 266 Mbit/s, 530 Mbit/s, and 1 Gbits/s) and on three types of both electrical and optical media. Data is transferred using a set of hierarchical functions. The lowest level (FC-0) defines the physical link in the system, including the fibre, connectors, optical and electrical parameters for a variety of data rates. At the FC-0 level, an electrical function 232 is capable of transmitting data at 1862 Mbps and an optical function 234 is capable of transmitting data at a Gigabit rate.

Detailed Description Text (11):
The Signaling Protocol (FC-2) level serves as the transport mechanism of Fibre Channel. In a framing protocol/flow control function 238, the framing rules of the data to be transferred between ports, the different mechanisms for controlling the three service classes and processes for managing the sequence of a data transfer are defined by FC-2.

Detailed Description Text (21):
FC-4, the highest level in the FC structure defines the application interfaces that can execute over Fibre Channel. It specifies the mapping rules of upper layer protocols using the FC levels below. Fibre Channel is equally adept at transporting both network and channel information and allows both protocol types to be concurrently transported over the same physical interface using a channel function 242 and a network function 244. The following network and channel protocols may be supported: Small Computer System Interface (SCSI); Intelligent Peripheral Interface (IPI); High Performance Parallel Interface (HIPPI) Framing Protocol; Internet Protocol (IP); ATM Adaptation Layer for computer data (AAL5); Link Encapsulation (FC-LE); Single Byte Command Code Set Mapping (SBCCS); and IEEE 802.2

Detailed Description Text (22):
A virtual volume management protocol (VVMP) 250 may exist above the FC-4 layer. The VVMP protocol may operate either on Fibre Channel or a LAN/WAN connection. The VVMP protocol can have features similar to LAN/WAN features such as NHRP, Shortest Path, SNMP query, and discovery, among others. Additionally, IVVM (Intelligent Virtual Volume Manager) can also work with a framework called Storex, available from Sun Microsystems, Inc. of Mountain View, Calif. A sample VVMP is shown in Appendix A.

Detailed Description Text (28):
The manager, server, or switch may communicate with each other or with the data storage devices using a number of suitable protocols, including Fibre Channel protocol, Internet Protocol (IP), Asynchronous Transfer Mode (ATM) protocol, or an in-band control protocol such as the Small Computer System Interface (SCSI) protocol.

Detailed Description Text (44):
7. Type--SCSI-I or II or III or Fibre Channel

h   e b   b g ee e f   e h   ef   b   e

Detailed Description Text (52):
7. Type--SCSI-I or II or III or Fibre Channel

Detailed Description Text (63):
3. SCSI or Fibre channel

Other Reference Publication (1):
Z. Meggyesi, "Fibre Channel Overview," downloaded from
www1.cern.ch/HSI/fcs/spec/overview.htm, Mar. 16, 1999, 9 pgs.

Other Reference Publication (2):
P. Massiglia, "Fibre Channel, Storage Area Networks, and Disk Array Systems,"
Adaptec, Inc. Apr. 13, 1998, 22 pgs.

CLAIMS:

9. The method of claim 5, wherein the manager communicates with each data storage
device using a Fibre Channel protocol.

17. The method of claim 16, wherein the server communicates using a Fibre Channel
protocol.

19. The method of claim 18, wherein the switch communicates using a Fibre Channel
protocol.

35. The system of claim 33, wherein the switch is a Fibre Channel switch.

| Full | Title | Citation | Front | Review | Classification | Date | Reference | Sequences | Attachments | Claims | KWIC | Draw De |

| Clear | Generate Collection | Print | Fwd Refs | Bkwd Refs | Generate OACS |

Term	Documents
FIBRE	35062
FIBRES	28227
FIBER	262175
FIBERS	216421
CHANNEL	520811
CHANNELS	289475
(2 AND (FIBRE ADJ1 CHANNEL)).USPT.	16
(L2 AND FIBRE ADJ1 CHANNEL).USPT.	16

Display Format: KWIC   Change Format

Previous Page   Next Page   Go to Doc#

h   e b   b g ee e f   e   h   ef   b   e